NAVAL POSTGRADUATE SCHOOL
Monterey, California

DISSERTATION

A CASCADE APPROACH FOR STAIRCASE LINEAR PROGRAMS WITH AN APPLICATION TO AIR FORCE MOBILITY OPTIMIZATION

by

Steven F. Baker

June 1997

Thesis Advisor: Richard E. Rosenthal

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1997	Doctoral Dissertation

4. TITLE AND SUBTITLE
A Cascade Approach for Staircase Linear Programs with an Application to Air Force Mobility Optimization

5. FUNDING NUMBERS
N 94063

6. AUTHOR(S)
Baker, Steven F.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

We develop a method to approximately solve a large staircase linear program that optimizes decisions over time. Also developed is a method to bound that approximation's error. A feasible solution is derived by a *proximal cascade*, which sequentially considers overlapping subsets of the model's time periods, or other ordinally defined set. In turn, we bound the cascade's deviation from the optimal objective value by a *Lagrangian cascade* which penalizes infeasibility by incorporating dual information provided by the proximal cascade solution. When tested on a large temporal LP developed for US Air Force mobility planners, we often observe gaps between the approximation and bound of less than 10 percent, and save as much as 80 percent of the time required to solve the original problem. We also address methods to reduce the gap, including constraint extension of the Lagrangian cascade, as well as exploitation of dual multipliers within the proximal cascade.

14. SUBJECT TERMS
Optimization, Linear Programming, Cascade, Rolling Horizon, USAF, Mobility Modeling

15. NUMBER OF PAGES
163

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

Approved for public release; distribution is unlimited

A CASCADE APPROACH FOR STAIRCASE LINEAR PROGRAMS WITH AN APPLICATION TO AIR FORCE MOBILITY OPTIMIZATION

Steven F. Baker
Major, United States Air Force
B.S., US Air Force Academy, 1981
M.S., Air Force Institute of Technology, 1991

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL

June, 1997

Author: _____

Steven F. Baker

Approved by: _____

Richard E. Rosenthal
Professor of Operations Research
Dissertation Supervisor

_____

Gerald G. Brown
Professor of Operations Research

_____

David P. Morton
Assistant Professor of Operations Research
The University of Texas at Austin

_____

Craig M. Rasmussen
Assistant Professor of Mathematics

_____

R. Kevin Wood
Associate Professor of Operations Research

Approved by: _____

Frank C. Petho, Chair, Department of Operations Research

Approved by: _____

Maurice D. Weir, Associate Provost for Instruction

ABSTRACT

We develop a method to approximately solve a large staircase linear program that optimizes decisions over time. Also developed is a method to bound that approximation's error. A feasible solution is derived by a *proximal cascade,* which sequentially considers overlapping subsets of the model's time periods, or other ordinally defined set. In turn, we bound the cascade's deviation from the optimal objective value by a *Lagrangian cascade,* which penalizes infeasibility by incorporating dual information provided by the proximal cascade solution. When tested on a large temporal LP developed for US Air Force mobility planners, we often observe gaps between the approximation and bound of less than 10 percent, and save as much as 80 percent of the time required to solve the original problem. We also address methods to reduce the gap, including constraint extension of the Lagrangian cascade, as well as exploitation of dual multipliers within the proximal cascade.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This research involved a lot of effort and patience from many individuals. I'd like to recognize a few of these people specifically:

- Professor Richard Rosenthal, who was much more than a dissertation advisor. He showed me an integrated view of what an academician, researcher and consultant must be. I will value his mentorship for my entire career, and his friendship for the rest of my life.
- Professor David Morton, who provided many of the key ideas for my (our) work. Without his contributions, I might still be looking for a topic.
- Professor Gerald Brown, who provided the key insight as to how to show the value of this research to a wide audience.
- Professor Craig Rasmussen, who inspired confidence and desire to learn by his classroom teaching and tireless tutoring.
- Professor Kevin Wood, who's technical writing skills have made this document far more readable and pertinent.

I'd also like to thank CDR Tom Halwachs for his willingness to provide computer assistance whenever asked, and Ms. Laura Melody for her valuable work on scenario generation.

Finally, my wife Donna, and children Kelly and Stacy, deserve considerable credit for putting up with a grumpy, preoccupied, and domineering beast for the past three years. I will try to reform!

EXECUTIVE SUMMARY

This dissertation develops techniques to solve very large instances of a linear program that optimizes US Air Force (USAF) strategic and tactical airlift for regional contingencies. Until recently, simulation and spreadsheet models were used for airlift analysis because sufficiently detailed optimization models were intractable due to their size. In order to facilitate the use of highly detailed mobility optimizations, we develop the *proximal cascade,* which approximates the solution of large linear programs that involve decisions over time, location, or priority. We also develop the *Lagrangian cascade,* which quantitatively assesses the approximation's accuracy.

A linear program may be approximated by the proximal cascade when a model's decisions directly affect only proximal decisions, *i.e.,* those that are closely related by some attribute such as time. A proximal cascade first considers only the earliest decision periods of a model, and then cascades forward in time to consider the decisions of later periods. The number of periods considered by each stage, or cascade subproblem, is often limited by the computational power available. Alternatively, subproblem size can be determined by the level of future uncertainty encountered in the system being modelled.

We assess the accuracy of the proximal cascade by the Lagrangian cascade. Lagrangian cascades also consist of proximally related subproblems, made separable by not enforcing resource limitations that involve the time periods of multiple subproblems. Instead of explicit enforcement, these resources express their scarcity by charging a consumption penalty, similar to the ones used in the proximal cascade.

A proximal cascade solution enforces all resource limitations in one or more subproblems. Therefore, its solution does not violate any of the assumptions made by the model formulation. However, its solution may not be the best possible, because it is encumbered by not being able to consider all periods at once. Conversely, a Lagrangian cascade may provide a solution that violates resource limitations, but is more economical by some objective cost measure than a fully constrained solution. Therefore, the two cascade objective costs bound the optimal objective cost, which is the cost when all periods are solved at once.

The two cascades are also related by consumption penalties. All linear programs yield as part of their solution, marginal values for all of the constraining resources. Marginal values from the proximal cascade are used in the Lagrangian cascade as appropriate penalty levels for consumption of the resources whose limits are not enforced. This circumvents a long search for the appropriate penalties that frequently hamper similar relaxation-penalty methods.

The Naval Postgraduate School/RAND Mobility Optimizer (NRMO) provides an excellent opportunity to test proximal and Lagrangian cascades. This model is the latest in an evolution of linear programing optimization models that address the increased interest in airlift mobility as a result of the Gulf War. NRMO optimizes decisions involving numerous aspects of a deployment, including strategic airlift mission and crew assignments, aerial refueling missions, intra-theater deliveries, and recovery options. Consequently, the size of this model can be huge.

When tested on several NRMO scenarios, the percentage gap between the proximal and Lagrangian cascade objective costs is often within 10%. In other words, the cost of the proximal cascade solution is within 10% of the optimal solution cost. Computation times vary, but can take as little as 20% of the time required to solve all periods at once

Cascades provide a useful approximation and bounding strategy for linear programs that exhibit a proximal decision structure. The method permits solution of model scenarios that are much larger than are otherwise possible, and has applicability to a linear program currently in use by the USAF.

I. INTRODUCTION

Large mathematical programs often require indirect solution methods that exploit the problem's structure. When the variables and constraints of a mathematical program can be ordered by some attribute such that all variables within any constraint are ordinally proximate, that mathematical program can be characterized as a "staircase model." The purpose of this research is to formalize a heuristic that exploits the structure of staircase linear programs, and provide a bound for that heuristic's accuracy. Once developed, we test the heuristic and bound on a large linear program used by the US Air Force (USAF) for air mobility analyses.

Staircase linear optimization models are widely used in many areas such as scheduling, where decisions of a given time period directly affect only the decisions of proximal time periods. The success of linear and integer programming (LP and IP, respectively) in aiding schedulers is well known. These models frequently consider a large but finite *solution horizon* [Walker and Dell, 1995], which is the number of time periods included in a scheduling model. Unfortunately, these models are limited by temporal considerations in at least two ways: 1) a distant solution horizon may make gathering accurate data for the latter periods problematic, and 2) a sufficiently large solution horizon may produce a model that is too large to solve. Not surprisingly, a human scheduler faces the same difficulties, namely reconciling the increasing number of options with decreasing certainty as the solution horizon grows. For either the human scheduler or the optimization model, perhaps the most straightforward way of dealing with the difficulties presented by a large problem is to focus on a subset of the problem's time periods, and then move forward to a new subset. This temporal *myopia*, or inability to see the full problem at any one point, may result in a suboptimal solution, but can make the problem simple enough to solve. Moreover, a model that is used to mimic scheduling, but not produce schedules, may be best if it can incorporate the realism of myopic scheduling. For example, when selecting fleet size or infrastructure in order to maximize a delivery system's effectiveness, a model should not unduly anticipate

delivery requirements far into the future. Thus, myopia is a desirable model characteristic whenever perfect foresight is unwarranted.

Modeling myopia is acceptable and realistic provided the resource commitments (constraints) initiated by decisions are short relative to the solution horizon. In an LP, this constraint-enforced linkage of a time period's variables only to nearby periods resembles a staircase along the main diagonal of the constraint coefficient matrix. The rest of the coefficient matrix is relatively sparse, since variables associated with the early time periods rarely appear in constraints corresponding to the later time periods. Thus, temporarily ignoring the variables and constraints associated with later periods may have only a small effect on the "early" portion of the solution, since most of the constraints for those periods are left intact.

Once a schedule is produced for a limited number of early periods, the earliest decisions are fixed, and the model "cascades" forward in order to solve for a later set of periods. Webster's [1993, p. 345] defines a cascade as:

> "A succession of stages (as in a process or in the arrangement of the parts of an apparatus) in which each stage derives from or acts, sometimes cumulatively, upon the product or output of the preceding."

In a mathematical program, a cascade implies generating a feasible solution by solving only a subset of a problem's constraints and variables, and then moving to a new subset corresponding to later time periods. Each of these cascade subproblems should re-solve a portion of the previous subproblem in order to minimize the end effects caused by the temporal limitation. This method of approximating an LP solution was first suggested by Charnes and Cooper [1961, pp. 370-388], and is often used to truncate problems with a theoretically infinite number of time periods. In contrast, a goal of this research is to implement a *proximal cascade* heuristic; a heuristic that sequentially selects and solves portions of a model whose variables and constraints are finitely indexed by an ordinally defined set such as time periods.

The closeness of a proximal cascade approximation to the overall LP solution is dependent on many scenario-specific factors. In order to supplement the proximal cascade approximation, this research also develops an optimistic bound on the LPs solution value

by exploiting information derived from the proximal cascade. When ordered by a time index, staircase LPs may have constraints that link only proximal time periods; relaxing the constraints associated with certain time periods can decouple a large problem into several subproblems. Lagrangian relaxation has long been used for this; it discourages violation of relaxed constraints through penalties. The Lagrangian penalty is applied to a series of separable problems, and an optimistic bound for the solution's objective value is derived. Unfortunately, finding the correct penalty values for relaxed constraints is often as difficult as solving the problem without the relaxation. However, this research shows that reasonable penalties for the relaxed constraints are readily available from the "shadow prices" of a proximal cascade solution. A *Lagrangian cascade* produces a bound on the LP solution by incorporating the proximal cascade penalties into a series of decoupled subproblems. When combined with the proximal cascade approximation, the size of the gap between the two values gives a quantitative assessment of proximal cascade accuracy.

Once developed, we demonstrate proximal and Lagrangian cascades on a large LP currently in use by USAF analysts for mobility planning. This model, under development concurrent with the cascade research, defies the long held opinion among many Air Force planners that LPs with sufficient detail to model the underlying mobility system are currently intractable due to their size. Proximal and Lagrangian cascades provide a methodology by which to allay that criticism, and are examined in this research using instances of the mobility LP.

A cascade can be used on a wide variety of problems for several different reasons, and may often be improved by altering problem structure or further exploiting dual information. We complete this research by examining how cascade performance on general LPs can be predicted and enhanced.

A. LITERATURE REVIEW

The topics germane to the research include decomposition of large LPs, Lagrangian relaxation, time-based or proximal methods, and military mobility optimization. While there is a wealth of literature on decomposition and Lagrangian relaxation, proximal meth-

ods and military mobility optimization are sparsely documented. Below is a summary of the literature.

1. Decomposition

The notion of incorporating dual information to decompose large linear programs into smaller, structured LPs originated with Dantzig and Wolfe [1960], and Benders [1962]. Both methods rely on passing primal and dual information between a *master* problem, which addresses the original problem in a simplified form, and one or more *subproblems*, which address portions of the problem in detail. These subproblems often exhibit a computationally exploitable structure.

In Dantzig-Wolfe decomposition, the subproblems use dual prices from the master problem in order to derive new variables for the master problem that will price favorably in subsequent iterations. In turn, the master problem takes a convex combination of these new variables to produce a feasible solution to the overall problem, as well as new resource prices for the subproblems. The method converges when the subproblems cannot find a variable to price favorably in the master problem.

Benders' decomposition of an LP is similar in many respects to Dantzig-Wolfe decomposition since the Benders' master problem is the dual of the Dantzig-Wolfe master problem. Consequently, instead of solving the master problem with a subset of variables (as in Dantzig-Wolfe), the Benders' master problem solves a subset of cuts derived from a reformulation of original constraints, but in the dual. In turn, the Benders' decomposition subproblems use the current master problem solution to produce a violated constraint to be appended to the next master problem iteration. The method has converged when the subproblem solution can no longer find a violated constraint.

Geoffrion and Graves [1974] use Benders' decomposition to reduce a mixed integer, multi-commodity transportation problem into separable single commodity problems. In this formulation, the master problem dictates the configuration of the integer variables based on cost information from the subproblems, while the subproblems determine the flows based on the network provided by the master problem. Brown, Graves, and Honczarenko [1987]

extend this technique using elastic constraints to insure feasibility as well as improve convergence.

Decomposition has also been applied to staircase linear programs by Glassey [1973] as well as Ho and Manne [1974]. Their method repeatedly applies the Dantzig-Wolfe technique to succeeding (or preceding) levels of a staircase LP, forming a "nested" decomposition. Each new staircase level acts as the next subproblem, which feeds back pricing information to its master (the previous staircase level), while sending variable levels forward to the next staircase level. A staircase LP can also be decomposed by Benders' method, as shown by Van Slyke and Wets [1969] for two-stage stochastic programming, and later by Birge [1985] for multi-stage stochastic programming.

Although not a decomposition technique, the solution of large-scale LPs can also be approximated by aggregation of time periods until a problem of workable size is derived. Zipkin [1980] describes a methodology for bounding the error incurred by such aggregation in some problems. Although the idea has merit for large models, and has been used to solve problems similar to the one described in this research, it has no direct applicability to proximal and Lagrangian cascades.

2. Lagrangian Relaxation

Lagrangian relaxation is used in many optimization applications, including vehicle routing, scheduling, and network design problems [*e.g.*, Ahuja, Magnanti, and Orlin, 1993, pp. 620-635]. Common to these methods is a search for accurate Lagrangian penalties of the relaxed constraints, which has proved the most difficult aspect of the overall method. Parker and Rardin [1988, pp. 205-237], as well as Bazaraa, Sherali, and Shetty [1993, pp. 199-231] give a summary of the search techniques. A Lagrangian cascade requires none of these techniques, since the Lagrangian penalties are a by-product of the proximal cascade. However, further tightening of the Lagrangian cascade bound could benefit from multiplier search techniques. This remains a subject for further research.

3. Time-Based, or Proximal Methods

The use of temporally progressing solution strategies in optimization is of two varieties; *solution cascading* and *forward optimization.* Brown, Graves, and Ronen [1987] implement solution cascading by solving successive portions of a model's time periods in order to produce an advanced basis. For example, a problem with 15 time periods is split into three smaller problems, each considering only rows and columns indexed with periods 1-5, 6-10, and 11-15, respectively. The optimal solutions of these subproblems are then used to suggest columns that price favorably, as well as produce an advanced, or "crash" basis for the original problem. With this "head start," optimality may ensue in fewer iterations. Jayakumar and Ramasesh [1994] demonstrate the computational savings of solution cascading on a number of test problems.

Forward optimization as outlined by Morton [1981] involves solving successively longer (more time periods) problems until a *decision horizon* is reached. A decision horizon is a point beyond which solving larger problems will not alter the decisions of the first time period. This method shows that (for some problems) an optimal solution can be reached by solving a succession of small LPs, and recording the values within each as optimal. Aronson *et al.* [1985] develop and test this idea for certain classes of problems, notably from the area of manufacturing. Production scheduling problems where time periods are linked only by inventory level exhibit natural decision horizons just after periods of maximum demand. At these points in time, inventories are exhausted, effectively restarting the production schedule. Thus, forward optimization is appealing when solving certain classes of problems, but does not offer general applicability.

Manne [1970] offers related work on limiting the temporal horizon of an LP. His research provides sufficient conditions for optimality when truncating infinite horizon LPs whose coefficients do not change in the latter periods. Walker [1995] extends this idea to bound the error produced by truncating infinite horizon LPs prior to the point where Manne showed equivalency between finite and infinite horizon problems. Unfortunately, the infinite horizon method requires an invariant constraint structure beyond a specified time period, which does not occur in all staircase problems. There is no body of literature

on the solution and bounding of large, but still finite, LPs by a proximal cascade, which successively solves portions of a non-homogeneous staircase LP in order to approximate an otherwise intractable problem.

4. Military Mobility Optimization

Dantzig and Fulkerson [1954] offered the first application of mathematical programming to time-dynamic military transportation problems. Their work scheduling US Navy tankers was seminal for military logistics optimization as well as time-dynamic network transportation problems.

Until recently, the computational demands of LP in modeling large-scale Air Force contingency deployments allowed an insufficient level of detail for many analyses. Consequently, simulation was the method of choice for analyzing fleet mix and infrastructure requirements of such a deployment. Wing *et al.* [1991] developed a time-dynamic LP as a response to the Mobility Requirements Study mandated by the National Defense Authorization Act of 1991. Yost [1994] continued the integration of LP into the mobility modeling arena with the development of THRUPUT in 1994, which offered a detailed routing structure, but was temporally static. Concurrent with Yost's work, the RAND Corporation developed CONOP [Killingsworth and Melody, 1994], which also focused on airlift, but initially examined the efficacy of aerial refueling of airlifters in a contingency. Lim [1994], Morton, Rosenthal, and Lim [1995], and Rosenthal *et al.* [1996] extended THRUPUT with the development of THRUPUT II, which incorporated the multiple time periods into Yost's work. Subsequently, RAND's CONOP model and THRUPUT II were merged into the Naval Postgraduate School/RAND (NPS/RAND) Mobility Optimizer [Rosenthal *et al.*, 1997], which is the case study considered in this dissertation.

B. EXPLANATION OF TERMS

Several key terms have a specific meaning in this research. A comprehensive list follows:

Monolith: A formulation a linear program. Many definitions that follow consider portions, or subsets of the monolith.

Row: A constraint of the monolith, defined by its indices, technological coefficients, sense ($\leq$, $=$, or $\geq$), and right hand side.

Column: A variable of the monolith, defined by its indices, coefficients, and bounds.

Association: If a row and column intersect with a nonzero technological coefficient, then they are said to be associated.

Cascade index set: A scalar attribute assigned to each row and column. The scalar may be an index, or a distinguished null index (conventionally zero) when the assignment of a specific scalar is inappropriate (this occurs if a row or column has no corresponding cascade index). The idea is to assign non-null scalars that express a relation or proximity among rows and columns with identical or nearly identical non-null indices.

Active Index Set: A distinguished subset of contiguous cascade indices and the null index.

Linkage: A row associated with columns endowed with distinct cascade indices creates a linkage between the indices.

Active Row: Row endowed with an active cascade index.

Lagrange Row: Row other than an active row represented only by its Lagrangian relaxation (referred to as "Lagrange-relaxed").

Relaxed Row: Row that is neither active nor Lagrange-relaxed.

Active Column: Column endowed with an active cascade index.

Fixed Column: Not an active column, but endowed with some value which may influence its associated rows. A fixed column's value equals its level when made inactive, or zero if the column has never been active.

Subproblem: Active rows and columns, where the objective may include terms contributed by Lagrange-relaxed rows and fixed columns, and the right hand sides may be influenced by fixed columns.

Cascade: A sequence of subproblems. The motive for using a cascade is to indirectly assemble an acceptable answer to the monolith with less effort than an outright direct attempt at solution. A cascade may, or may not, culminate in a strictly feasible, optimal solution to the monolith. However, prescribing a useful solution for the problem from which the model monolith derives is the goal and guide.

Width: The range of non-null cascade indices active in, say, a subproblem or a row.

Overlap: The range of the subset of non-null cascade indices in common between, say, two subproblems or two rows.

Proximal Cascade: A non-separating sequence of subproblems, each of which has width intentionally constrained to represent some limited effective planning horizon less that the total number of cascade indices. A proximal cascade may be used to enhance computational tractability, or to temper unrealistic omniscience in a model monolith that represents a problem that would in reality be dealt with myopically.

Lagrangian Cascade: A separating sequence of subproblems defined by exhaustive partition of the cascade index set and rendered disjoint by Lagrangian relaxation of any row associated with two or more subproblems.

Gap (absolute): The absolute value of the difference between the proximal and Lagrangian cascade objective function values.

Solution quality: The inverse of the absolute gap between the monolith objective function value and the (proximal or Lagrangian) cascade objective function value. Solution quality equals infinity when this absolute gap is zero; solution quality equals zero when the cascade is infeasible or unbounded.

Gap (relative): The absolute gap divided by the absolute value of its more favorable constituent value (the lower value for a minimization problem). Relative gaps are assumed herein.

C. OVERVIEW

Chapter II develops proximal and Lagrangian cascade theory. The context used is a production setting, with the time index serving as the cascade index set. Rather than use the simplest staircase model, this chapter incorporates formulation complexities that ease the transition into the case study. Foremost among these characteristics is "elastic demand," which serves the dual purpose of supporting the case study, as well as demonstrating the flexibility of a cascade beyond simple staircase models.

Chapter III outlines the implementation of proximal and Lagrangian cascades by presenting a discussion and pseudocode of each. The remainder of the chapter considers the ramifications of heuristic parameter selection on problem feasibility and solution quality.

Chapter IV describes the USAF mobility model under development at NPS, and gives specific formulations for the proximal and Lagrangian cascades. Much of this chapter reconciles the theoretical development with the inevitable complexity of a "real world" model. The rest of the chapter describes cascade performance on a number of problem instances.

Chapter V generalizes a cascade to an arbitrary model, and offers a method to assess whether a model cascade might produce a feasible result of good quality. The chapter also discusses what conditions suggest whether or not a model cascade is warranted. Finally, the chapter considers how additional dual and primal information may be incorporated to

improve cascade solution quality. The most interesting of these methods uses an approach, similar to Benders' decomposition, to iteratively reduce the cascade gap.

The research is concluded in Chapter VI, which summarizes the theoretical and computational results. Chapter VI also suggests future opportunities for cascade research.

II. CASCADE THEORETICAL DEVELOPMENT

This chapter introduces and develops proximal and Lagrangian cascades. Although there are many variations of staircase problems, this research principally considers a scheduling problem that is formulated as an elastic-demand staircase model. To that end, this chapter first derives the single-commodity, elastic-demand staircase model from a familiar production-scheduling LP. Subsequently, we use that model to introduce the proximal cascade by segmenting it into smaller pieces. This segmentation produces a series of smaller problems that jointly approximate the monolith. Next, we develop proximal cascade theory for a multi-commodity elastic-demand staircase model, as well as for a generalized staircase model without elastic demands. Finally, the chapter introduces Lagrangian cascade theory, which provides a bound on the monolith's optimal objective value.

A. SINGLE-COMMODITY ELASTIC-DEMAND STAIRCASE LP

Preliminary use of a simple model is warranted. Although the case study for this research focuses on a military mobility scenario, the most familiar model setting involves scheduling of manufacturing resources. Consider the following single-commodity production-scheduling LP with elastic demands, multiple period lead times, and no inventory costs. In this case, assume the lead time is two periods, so production started in period t consumes resources in periods t and $t+1$, and can meet demand as early as period $t+1$:

INDICES

t	Time periods ($t = 1, 2...T$)

DATA

d_t	Demand in period t ($d_1 = 0$)
s_t	Production resource available in period t ($s_t > 0$)
$a_{tt'}$	Production resource consumption in period t per unit of production started in period t'. (In general, $a_{tt'}$ is not restricted to be positive unless specified.)

VARIABLES

X_t	production started in period t ($X_T = 0$, due to lead time)
I_t	inventory at the end of period t ($I_1 = 0,\ I_T = 0$)
P_t	elastic variable for unsatisfied demand in period t

FORMULATION

$$\min \sum_{t=1}^{T} P_t$$

$$\begin{aligned} s.t. \quad & X_{t-1} - I_t + I_{t-1} + P_t = d_t \qquad 1 < t \leq T \\ & a_{11} X_1 \leq s_1 \\ & a_{t,t-1} X_{t-1} + a_{tt} X_t \leq s_t \qquad 1 < t < T \\ & a_{T,T-1} X_{T-1} \leq s_T \\ & X_t, I_t, P_t \geq 0 \qquad \forall\, t \end{aligned}$$

Assuming that all the demand is in the last period, *i.e.*, $d_t = 0\ \forall t < T$, the inventory variables may be eliminated by noting that $I_2 = X_1$, and $I_t = X_{t-1} - I_{t-1}$ (see Johnson and Montgomery [1974, pp. 197-199] for a detailed discussion). Rewriting P_t as P, the problem may be reformulated as

$$(A) \qquad Z^A = \min\ P$$

$$\begin{aligned} s.t. \quad & \sum_{t=1}^{T-1} X_t + P = d && (A.1) \\ & a_{11} X_1 \leq s_1 && (A.2) \\ & a_{t,t-1} X_{t-1} + a_{tt} X_t \leq s_t \qquad 1 < t < T && (A.3) \\ & a_{T,T-1} X_{T-1} \leq s_T && (A.4) \\ & P \geq 0, \quad X_t \geq 0 \ \ \forall\, t. && (A.5) \end{aligned}$$

The remainder of this section assumes A has a finite optimal solution $X_t^A,\ 1 \leq t < T$ (throughout this document, a superscript on a variable denotes the variable's optimal value in the superscripted problem).

This simple problem offers a notationally straightforward way to demonstrate a cascade on a staircase problem, and it incorporates the additional richness of a complication such as elastic demand.

1. Segmented Approximation

A segmented approximation is a restricted version of a proximal cascade and it provides a good introduction. Problem A is made separable and approximated by removing, or setting equal to zero, column, $X_{\tau+1}$ for some value of τ between 1 and $T-1$. The following two maximization problems serve this purpose by separating a restricted version of A into two subproblems, one optimizing periods 1 to τ, and the other optimizing periods $\tau+1$ to T. The objective of each is to maximize production, rather than minimize penalties (We address the objective function sense in greater detail shortly).

Define the subproblem *SA1* (with the solution $X_t^{sa1}, 1 \leq t \leq \tau$, and the solution value Z^{sa1}) by

$$(SA1) \qquad Z^{sa1} = \max \sum_{t=1}^{\tau} X_t$$

$$s.t. \quad a_{11}X_1 \leq s_1 \tag{SA1.1}$$

$$a_{t,t-1}X_{t-1} + a_{tt}X_t \leq s_t \qquad 1 < t \leq \tau \tag{SA1.2}$$

$$a_{\tau+1,\tau}X_\tau \leq s_{\tau+1} \tag{SA1.3}$$

$$X_t \geq 0 \quad 1 \leq t \leq \tau. \tag{SA1.4}$$

Similarly, define the subproblem *SA2* (with the solution $X_t^{sa2}, \tau+1 < t < T$, and the solution value Z^{sa2}) by

$$(SA2) \qquad Z^{sa2} = \max \sum_{t=\tau+2}^{T-1} X_t$$

$$s.t. \quad a_{\tau+2,\tau+2}X_{\tau+2} \leq s_{\tau+2} \tag{SA2.1}$$

$$a_{t,t-1}X_{t-1} + a_{tt}X_t \leq s_t \qquad \tau+2 < t < T \tag{SA2.2}$$

$$a_{T,T-1}X_{T-1} \leq s_T \tag{SA2.3}$$

$$X_t \geq 0 \quad \tau+1 < t < T. \tag{SA2.4}$$

Given these two subproblem values, a bound on the solution to A is readily available. Using the notation $[x]^+ = \max[0, x]$, the following proposition demonstrates this relationship.

Proposition 2.1 *If SA1 and SA2 have finite solutions,* $Z^A \leq [d - Z^{sa1} - Z^{sa2}]^+$.

Proof: Removing the column $X_{\tau+1}$ (fixing at 0) from A produces a restriction, but also hints at separability:

$$\begin{aligned} Z^A \quad & \leq \min \quad P \\ & s.t. \quad \sum_{t=1}^{\tau} X_t + \sum_{t=\tau+2}^{T-1} X_t + P = d \\ & \qquad (SA1.1), ..., (SA1.4) \\ & \qquad (SA2.1), ..., (SA2.4). \end{aligned}$$

Solving for P, and noting that $P \geq 0$, the above is restated as

$$Z^A \leq \left[\begin{array}{ll} \min & d - \sum_{t=1}^{\tau} X_t - \sum_{t=\tau+2}^{T-1} X_t \\ s.t. & (SA1.1), ..., (SA1.4), \ (SA2.1), ..., (SA2.4) \end{array} \right]^+.$$

The non-negative stipulation is an important aspect of the problem, since *SA1* and *SA2* do not restrict the variable sums to be less than d. Because the constraint structure is separable, the right side of this inequality may be rewritten as

$$\left[d - \left(\begin{array}{ll} \max & \sum_{t=1}^{\tau} X_t \\ s.t. & (SA.1.1), ..., (SA1.4) \end{array} \right) - \left(\begin{array}{ll} \max & \sum_{t=\tau+2}^{T-1} X_t \\ s.t. & (SA2.1), ..., (SA2.4) \end{array} \right) \right]^+,$$

or:

$$\left[d - Z^{sa1} - Z^{sa2} \right]^+.$$

Thus we have

$$Z^A \leq \left[d - Z^{sa1} - Z^{sa2} \right]^+.$$

□

Observe that the combined solutions to *SA1* and *SA2* are feasible to A in the absence of over-production, since restricting $X_{\tau+1}$ to zero makes A equivalent to *SA1* and *SA2*, which are feasible by assumption. Thus, a feasible approximation to A is produced by

solving small subproblems. Unfortunately, the bound may be weak when $X_{\tau+1}$ is positive in the optimal solution to the monolith. Moreover, the method does not work for more complex problems. These shortcomings can be fixed by a proximal cascade, shown below.

2. Proximal Cascade Approximation

The segmented approximation removes columns to produce separable subproblems, foregoing any potential objective function improvement from those columns. The proximal cascade partially redresses this disadvantage, and we will show that its objective value is bounded from above by the segmented approximation just presented. This approach also solves the problem in piece-wise fashion, but uses a sequential method that fixes column levels from the latter periods of preceding subproblems. In turn, those fixed levels are incorporated into successor subproblems, allowing an approximation by a cascade of subproblems. To demonstrate, assume that subproblem *SA1* has been solved, and that the column levels of periods $t < \tau$ are fixed. Since $X_{\tau-1}$ is the only column to *directly* influence periods τ and greater, the second subproblem may be rewritten to incorporate *SA1* using just the level of $X_{\tau-1}$. Towards that end, define problem *CA2* (with the solution X_t^{ca2}, $\tau \leq t \leq T-1$) by

$$Z^{ca2} = \sum_{t=1}^{\tau-1} X_t^{sa1} + \max \sum_{t=\tau}^{T-1} X_t$$

$$\begin{aligned}
s.t. \qquad a_{\tau\tau}X_\tau &\leq s_\tau - a_{\tau,\tau-1}X_{\tau-1}^{sal} && (CA2.1)\\
a_{t,t-1}X_{t-1} + a_{tt}X_t &\leq s_t \quad \tau < t < T && (CA2.2)\\
a_{T,T-1}X_{T-1} &\leq s_T && (CA2.3)\\
X_t &\geq 0 \quad \tau \leq t < T. && (CA2.4)
\end{aligned}$$

Additionally, let

$$Z^{cas} = \left[d - Z^{ca2}\right]^+.$$

This value is the proximal cascade approximation of problem A. The following proposition relates the solution value of A to its proximal and segmented approximations:

Proposition 2.2 *If SA1, SA2 and CA2 have finite solutions,*

$$Z^A \leq Z^{cas} \leq \left[d - Z^{sa1} - Z^{sa2}\right]^+.$$

Proof: To show the right-hand inequality, first note that removing column $X_{\tau+1}$ restricts *CA2* , and thus

$$\begin{aligned}
Z^{ca2} \geq \sum_{t=1}^{\tau-1} X_t^{sa1} + \max \quad & X_\tau + \sum_{t=\tau+2}^{T-1} X_t & & & & \\
s.t. \quad & a_{\tau\tau} X_\tau & & & \leq & s_\tau - a_{\tau,\tau-1} X_{\tau-1}^{sa1} & \\
& a_{\tau+1,\tau} X_\tau & & & \leq & s_{\tau+1} & \\
& & a_{\tau+2,\tau+2} X_{\tau+2} & & \leq & s_{\tau+2} & \\
& & a_{t,t-1} X_{t-1} & + a_{tt} X_t & \leq & s_t & \tau+2 < t < T \\
& & & a_{TT-1} X_{T-1} & \leq & s_T & \\
& & & X_t & \geq & 0 & \tau+1 < t < T
\end{aligned}$$

$$\begin{aligned}
= \sum_{t=1}^{\tau-1} X_t^{sa1} + \max \quad & X_\tau & & \\
s.t. \quad & a_{\tau\tau} X_\tau & \leq & s_\tau - a_{\tau,\tau-1} X_{\tau-1}^{sa1} \\
& a_{\tau+1,\tau} X_\tau & \leq & s_{\tau+1} \\
& X_\tau & \geq & 0
\end{aligned}$$

$$\begin{aligned}
+ \max \quad & \sum_{t=\tau+2}^{T-1} X_t & & & & \\
s.t. \quad & a_{\tau+2,\tau+2} X_{\tau+2} & & \leq & s_{\tau+2} & \\
& a_{t,t-1} X_{t-1} & + a_{tt} X_t & \leq & s_t & \tau+2 < t < T \\
& & a_{TT-1} X_{T-1} & \leq & s_T & \\
& & X_t & \geq & 0 & \tau+1 < t < T
\end{aligned}$$

$$= \quad Z^{sa1} + Z^{sa2}.$$

Rearranging terms yields the desired result:

$$-Z^{ca2} \leq -Z^{sa1} - Z^{sa2}$$

$$Z^{cas} = \left[d - Z^{ca2}\right]^+ \leq \left[d - Z^{sa1} - Z^{sa2}\right]^+.$$

This is the right-hand inequality. Note also that $X_t^{sa1}, 1 < t < \tau$, and $X_\tau^{ca2}, X_t^{ca2}, \tau + 1 < t < T$ is feasible to A, since *SA1* and *CA2* jointly enforce the constraints of A.

To show the left-hand inequality, note that setting $X_{\tau-1} = X^{sal}_{\tau-1}$ restricts problem A. Stated in the form derived at the beginning of the chapter:

$$
\begin{aligned}
Z^A \;\leq\; & \left[\begin{array}{c} d - \max \sum_{t=1}^{T-1} X_t \\ s.t. \quad (A.2), ..., (A.4), (SA1.4), (CA2.4), \; X_\tau = X^{sal}_\tau \end{array} \right]^+ \\
= \; & \left[\begin{array}{c} d - X^{sa1}_\tau - \max \sum_{t=1}^{\tau-2} X_t + \sum_{t=\tau}^{T-1} X_t \\ s.t. \; (SA1.1), \; (SA1.4), \\ a_{t,t-1} X_{t-1} + a_{tt} X_t \leq s_t \quad 1 < t < \tau - 1 \\ a_{\tau-1,\tau-2} X_{\tau-2} \leq s_{\tau-1} - a_{\tau-1,\tau-1} X^{sal}_{\tau-1} \\ (CA2.1), ..., (CA2.4) \end{array} \right]^+ \\
= \; & \left[d - \sum_{t=1}^{\tau-1} X^{sal}_t - \sum_{t=\tau}^{T-1} X^{ca2}_t \right]^+ \\
& [d - Z^{ca2}]^+ \;\; = Z^{cas}.
\end{aligned}
$$

□

This result shows that a feasible approximation to problem A is obtained by proximal cascade. This approximation is no further from the optimal solution value than a segmented approximation of A, and it is better than the segmented approximation when production in period $\tau + 1$ is beneficial. The next section shows that the proximal cascade also provides a lower bound on the optimal objective value of A.

3. Lower Bound by Proximal Cascade

If the assumptions given to this point are supplemented with the non-negativity of $a_{tt'}$, a lower bound on the solution to A is available from the proximal cascade subproblems. Thus, for the cost of solving *SA1* and *CA2*, one obtains a feasible approximation of the solution to A, as well as an assessment of its quality.

To show this result, a preliminary lemma is required:

Lemma 2.3 *If* $a_{tt'} \geq 0 \;\; \forall t, t', \;\; Z^{sal} \geq \sum_{t=1}^{\tau} X^A_t$.

Proof:

$$
\begin{aligned}
Z^{sal} &= \max \quad \sum_{t=1}^{\tau} X_t \\
&\quad s.t. \quad (SA1.1), ..., (SA1.4) \\
&\geq \max \quad \sum_{t=1}^{\tau} X_t \\
&\quad s.t. \quad (SA1.1), (SA1.2), (SA1.4) \\
&\qquad a_{\tau+1,\tau} X_\tau \leq s_{\tau+1} - a_{\tau+1,\tau+1} X^A_{\tau+1} \\
&= \sum_{t=1}^{\tau} X^A_t .
\end{aligned}
$$

□

The equality holds because fixing $X_{\tau+1}$ to $X^A_{\tau+1}$ allows the remaining columns to take their optimal values from problem A. With this lemma (and the non-negativity assumption of $a_{tt'}$), the following proposition establishes that a lower bound derives from "double counting" the levels of columns that are active in both subproblems. In this case, X_τ is the "double counted" column, since it is active in both subproblems:

Proposition 2.4 *If* $a_{tt'} \geq 0 \ \ \forall t, t', \ \ [d - Z^{ca2} - X^{sal}_4]^+ \leq Z^A$

Proof: Reducing the right-hand side of the period $\tau + 1$ inequality from $s_{\tau+1}$ to $s_{\tau+1} - a_{\tau+1,\tau} X^A_\tau$ is a restriction of subproblem *CA2*. Stipulating $X_\tau = 0$ further restricts *CA2*:

$$
\begin{aligned}
Z^{ca2} \geq \sum_{t=1}^{\tau-1} X^{sal}_t + \ & \max \quad \sum_{t=\tau}^{T-1} X_t \\
& s.t. \quad (CA2.1) \\
& a_{\tau+1,\tau} X_\tau + a_{\tau+1,\tau+1} X_{\tau+1} \leq s_{\tau+1} - a_{\tau+1,\tau} X^A_\tau \\
& \qquad a_{t,t-1} X_{t-1} + a_{t,t} X_t \leq s_t \qquad \tau + 2 \leq t < T \\
& \qquad\qquad X_\tau = 0 \\
& \qquad (CA2.3), (CA2.4)
\end{aligned}
$$

$$
\begin{aligned}
= \sum_{t=1}^{\tau-1} X^{sal}_t + \ & \max \quad \sum_{t=\tau+1}^{T-1} X_t \\
& s.t. \quad a_{\tau+1,\tau+1} X_{\tau+1} \leq s_{\tau+1} - a_{\tau+1,\tau} X^A_\tau \\
& \qquad a_{t,t-1} X_{t-1} + a_{t,t} X_t \leq s_t \quad \tau + 2 \leq t < T \\
& \qquad (CA2.3), (CA2.4)
\end{aligned}
$$

$$
= \sum_{t=1}^{\tau-1} X^{sal}_t + \sum_{t=\tau+1}^{T-1} X^A_t .
$$

The last equality holds because fixing X_τ to X_τ^A allows columns indexed by periods $\tau+1$ to period T to take their optimal values from problem A. Thus

$$Z^{ca2} + X_\tau^{sal} \geq \sum_{t=1}^{\tau-1} X_t^{sal} + X_\tau^{sal} + \sum_{t=\tau+1}^{T-1} X_t^A$$

$$= Z^{sal} + \sum_{t=\tau+1}^{T-1} X_t^A.$$

Combining this with lemma II.3, we have

$$Z^{ca2} + X_\tau^{sal} \geq \sum_{t=1}^{T-1} X_t^A,$$

or

$$-Z^{ca2} - X_\tau^{sal} \leq -\sum_{t=1}^{T-1} X_t^A.$$

Thus

$$\left[d - Z^{ca2} - X_\tau^{sal}\right]^+ \leq \left[d - \sum_{t=1}^{T-1} X_t^A\right]^+ = Z^A.$$

□

The results of this section use a very simple problem, but provide the groundwork for the remaining research. However, these results must be generalized to multiple commodities and other staircase problems before they become useful for the motivating problems of this dissertation. During the course of this development, we show that only the proximal cascade upper bound holds in a more general setting. Thus, a revised lower bound must be developed. That development, as well as the generalization of the proximal cascade, is the subject of the remainder of the chapter.

B. MULTI-COMMODITY ELASTIC-DEMAND STAIRCASE LP

Although the single-commodity elastic-demand staircase problem offers interesting results with respect to a proximal cascade, its usefulness is limited by the assumption of a single-commodity. These next two sections generalize proximal cascade results, first to the multi-commodity problem, then to more general staircase models.

1. Inapplicability of Segmented Bounds

Unlike its single-commodity counterpart, the multi-commodity elastic-demand staircase problem does not easily lend itself to segmented solution. Consider the following two-period problem with two commodities referenced by X and Y:

$$
\begin{array}{llccccccccc}
(A1) & Z^{A1} = \min & & & & & & +P_X & +P_Y & & \\
& s.t & X_1 & & +X_2 & & & +P_X & & = & 10 \\
& & & +Y_1 & & +Y_2 & & & +P_Y & = & 10 \\
& & 2X_1 & +Y_1 & & & & & & \leq & 6 \\
& & 2X_1 & +Y_1 & +2X_2 & +Y_2 & & & & \leq & 12 \\
& & & & +3X_2 & +Y_2 & & & & \leq & 6 \\
& & X_1, & Y_1, & X_2, & Y_2, & P_X, & & P_Y & \geq & 0,
\end{array}
$$

which has solution

$$Z^{A1} = 9,\ X_1^{A1} = 1,\ Y_1^{A1} = 4,\ X_2^{A1} = 0,\ Y_2^{A1} = 6,\ P_X^{A1} = 9,\ P_Y^{A1} = 0.$$

Also consider the subproblem

$$
\begin{array}{llccccccc}
(A2) & Z^{A2} = \max & X_1 & +Y_1 & +X_2 & +Y_2 & & & \\
& s.t & 2X_1 & +Y_1 & & & \leq & 6 & \\
& & 2X_1 & +Y_1 & +2X_2 & +Y_2 & \leq & 12 & \\
& & & & +3X_2 & +Y_2 & \leq & 6 & \\
& & X_1, & Y_1, & X_2, & Y_2, & \geq & 0, &
\end{array}
$$

which has solution

$$Z^{A2} = 12,\ X_1^{A2} = 0,\ X_2^{A2} = 0,\ Y_1^{A2} = 6,\ Y_2^{A2} = 6.$$

Direct extension of the segmented approach to the multi-commodity case requires the equivalence of $Z^{A1}$and $\left[d_X + d_Y - Z^{A2}\right]^+$. However, this does not hold here, since

$$Z^{A1} = 9 > 8 = \left[d_X + d_Y - Z^{A2}\right]^+.$$

The difficulty springs from overproducing the "easy" commodity Y and using it to offset under production of X, which consumes more resources. While it is possible to redress this shortcoming by retaining the original form of the demand rows, doing so eliminates the ability to show that the segmented problem is an upper bound to the proximal cascade solution. Consequently, the segmented approximation is not considered further.

2. The Multi-Commodity Proximal Cascade

Despite the inability to use the segmented solution as a bound on the proximal cascade, a proximal cascade provides an upper bound on the problem of interest, namely a multi-commodity elastic-demand staircase problem. Below is the general formulation of this problem, where each commodity has an allowable production time window of consecutive periods. This problem will serve as the monolith for the remainder of the chapter:

INDICES AND INDEX SETS

$i \in I$	Commodities
$t, t' \in T$	Time periods
$t \in T_i$	Allowable time periods for initiating production of i
$t' \in TS_t$	Periods of initiated production that consume resources in period t

DATA

d_i	Demand for commodity i, due when production begun in the last period of T_i is complete
h_i	Penalty per unit of not delivering commodity i
s_t	Production resources available at time t ($s_t \geq 0$)
$a_{itt'}$	Resource consumption in period t per unit of i begun in period t' Thus, $a_{itt'} = 0$ unless $t' \in TS_t$ (in general, $a_{itt'}$ is not restricted to be positive unless specified)

VARIABLES

X_{it}	Production of i begun in period t
P_i	Elastic variable for unsatisfied demand of commodity i

FORMULATION

$$
\begin{aligned}
(B) \quad Z^B = \min \quad & \sum_{i \in I} h_i P_i & & & \\
s.t. \quad & \sum_{t \in T_i} X_{it} + P_i = d_i & \forall i \in I \quad & (\alpha_i) & (B.1) \\
& \sum_{i \in I} \sum_{t' \in T_i \cap TS_t} a_{itt'} X_{it'} \leq s_t & \forall t \in T \quad & (\beta_t) & (B.2) \\
& X_{it}, P_i \geq 0 \ \ \forall i \in I \ , t \in T & & & (B.3)
\end{aligned}
$$

Assume B has a finite optimal solution, X_{it}^B, P_i^B $\forall i, t$.

Now consider N overlapping subsets of contiguous time periods within set T that suggest subproblems (Figures 1 and 2):

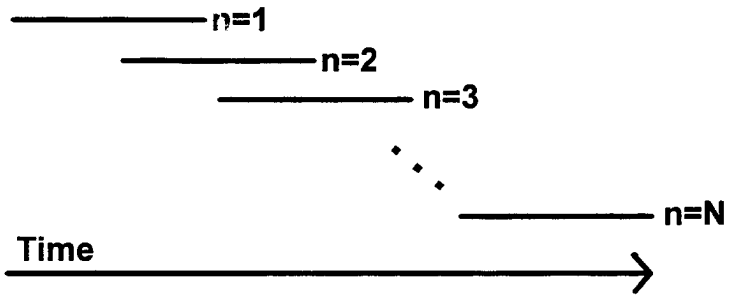

Figure 1. Sequence of subproblems forming a proximal cascade. The subproblems contain rows and columns indexed by overlapping subsets of active time periods.

Define the following:

$firstp^n$	First time period of subproblem n
$lastp^n$	Last time period of subproblem n
$caswid$	$\max_n [lastp^n - firstp^n] + 1$, the proximal cascade subproblem width
TC^n	$\{t \in T : firstp^n \leq t \leq lastp^n\}$, the active index set of subproblem n
m	$\max_t [\lvert TS_t \rvert] - 1$, the maximum resource utilization (staircase) overlap
v	$lastp^n - firstp^{n+1} + 1$, the number of time periods overlapping each subproblem (cascade overlap, assumed to be constant)
TF^n	$\{t \in T : firstp^n \leq t < firstp^{n+1}\}$ for $n < N$ $\{t \in T : firstp^n \leq t\}$, for $n = N$ the periods of TC^n, up to , but not including $firstp^{n+1}$.
NC	$n = \{1, ..., N\}$, the set of subproblems forming the proximal cascade

Note that there are two overlap parameters defined. Parameter m is the *staircase overlap*, and is a characteristic of the LP formulation. In contrast, v is the *cascade overlap*, and is a proximal cascade parameter. The next chapter discusses the ramifications of choosing v. However, v should be at least as large as m in to promote cascade feasibility.

The above definitions permit specification of N proximal cascade subproblems, CAS^n. Under the assumption that a finite optimal solution exists, let X_{it}^n, P_i^n solve

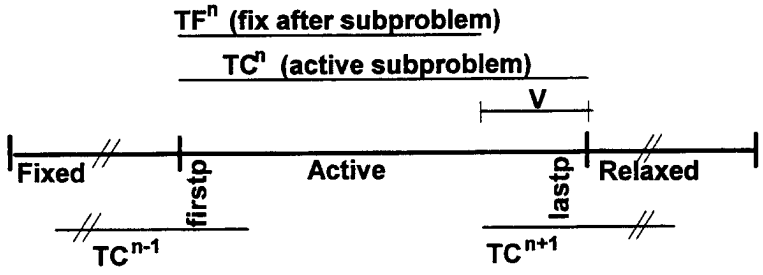

Figure 2. A single proximal cascade subproblem optimizes rows and columns indexed by the active set ($t \in TC^n$). Thus it re-optimizes rows and columns indexed by time periods active in the previous subproblem, $t \in TC^{n-1} \cap TC^n$. Rows of future time periods are relaxed, future columns are fixed at level 0. Columns of subproblem n that are not active in subproblem $n+1$ (indexed by $t \in TF^n$) are fixed at the end of n.

$$(CAS^n) \quad Z^n = \min \sum_{i \in I} h_i P_i$$

$$s.t. \sum_{t \in T_i \cap TC^n} X_{it} + P_i = d_i - \sum_{n' < n} \sum_{t \in T_i \cap TF^{n'}} X_{it}^{n'} \quad \forall i \in I \qquad (CAS^n.1)$$

$$\sum_{i \in I} \sum_{t' \in T_i \cap TS_t \cap TC^n} a_{itt'} X_{it'} \leq s_t - \sum_{n' < n} \sum_{i \in I} \sum_{t' \in T_i \cap TS_t \cap TF^{n'}} a_{itt'} X_{it'}^{n'} \quad \forall t \in TC^n \qquad (CAS^n.2)$$

$$X_{it}, P_i \geq 0 \quad \forall i \in I, t \in T_i \cap TC^n. \qquad (CAS^n.3)$$

The proximal cascade heuristic proceeds as follows (a detailed pseudocode is given in Chapter III):

For each $n \in NC$ {
  Define and solve subproblem CAS^n given above
  Fix the value of $X_{it}^n \ \forall i \in I, \ t \in TF^n$
}
Output proximal cascade solution: $X_{it}^n \ \forall i \in I, \ t \in TF^n, \ n \in NC,$ with value Z^N.

Each subproblem n activates all penalty columns and demand rows, but only the X_{it} columns for $t \in TC^n$. However, the subproblems have successively more fixed X_{it}^n values from previous subproblems. Thus, the demands of the last subproblem N are reduced by the solution values from TF^1 through TF^{N-1}. Similarly, staircase row right-hand-sides are reduced by fixed terms from previous subproblems.

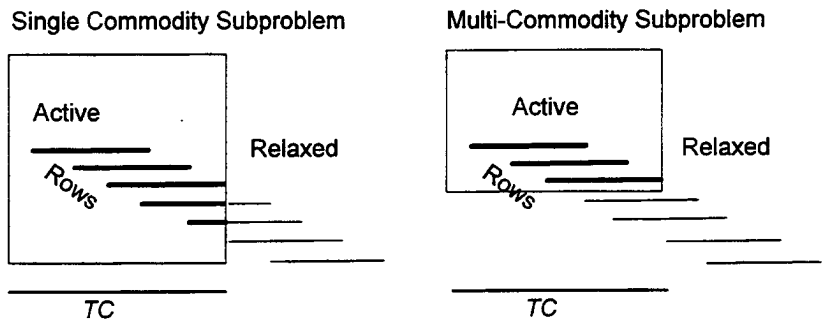

Figure 3. Unlike its single commodity counterpart, a multi-commodity proximal cascade subproblem excludes both columns *and* rows indexed by future time periods. These rows are required only for the segmented results of the previous section, and may cause infeasibility in more general problems.

In addition to multiple commodities and generalized notation, CAS^n differs slightly from the single-commodity subproblem *CA2*. Figure 3 illustrates that a subproblem CAS^n ending with period t activates the staircase rows only up to period t, while the single-commodity method activates all remaining rows associated with period t columns. Whereas these additional rows are useful when comparing to the segmented solution, they are not required in the general case, and may cause infeasibility or reduce solution quality of the cascade. For example, consider a model where $a_{itt'}$ is negative for the latter columns of a staircase row. Activating the row without those latter columns may force other associated columns to unnecessarily low values in order to maintain feasibility. Thus, any staircase row whose associated columns are not either active or fixed is not activated in CAS^n; each subproblem ends with the staircase rows indexed by the last period of TC^n.

The following theorem shows that the solution value of a cascade's final subproblem (Z^N) provides an upper bound on B.

Theorem 2.5 $Z^B \leq Z^N$.

Proof:

$$\begin{aligned} Z^B = \quad & \min \quad && \sum_{i \in I} h_i P_i \\ & s.t. && (B.1), (B.2), (B.3) \\ \leq \quad & \min \quad && \sum_{i \in I} h_i P_i \\ & s.t. && (B.1), (B.2), (B.3) \\ & && X_{it} = X_{it}^n \quad \forall n < N, t \in T_i \cap TF^n \\ = \quad & \min \quad && \sum_{i \in I} h_i P_i \\ & s.t. && (CAS^N.1), (CAS^N.2), (CAS^N.3), \\ = \quad & Z^N. \end{aligned}$$

The inequality holds since fixing a subset of X_{it} restricts the original problem.

□

Although similar to the proof given for the single-commodity problem, this proof restricts all column levels for $t <$*firstp*^{N} to their associated subproblem's value. However, fixing only the columns of the overlapping staircase periods (*firstp*$^n - m \leq t \leq$*firstp*^{n}, $\forall n \in NC$) gives the same result, since that restriction results in separable problems, namely $CAS^n, \forall n \in NC$.

The above proof shows that the proximal cascade solution provides an upper bound on Z^B. Additionally, the solution given by the cascade result ($X_{it}^n, \forall n \in N, i \in I,\ t \in TF^n$) is feasible to B, since the rows of B are enforced by the rows of $CAS^n\ \forall n \in NC$.

C. PROXIMAL CASCADES WITH BASIC STAIRCASE LPs

A proximal cascade is applicable to basic staircase models. This section extends the upper and lower proximal cascade bounds (developed for the single-commodity demand problem) to a simple staircase problem.

Consider problem S below. Parameter h_t is defined as the objective cost coefficient; otherwise the notation is the same as in problem B.

$$(S) \qquad Z^S = \max \sum_{t \in T} h_t X_t$$

$$s.t. \quad \sum_{t' \in TS_t} a_{tt'} X_{t'} \leq s_t \quad \forall t \in T \qquad (S.1)$$

$$X_t \geq 0 \qquad \forall t \in T \qquad (S.2)$$

Similarly (using the same notation as in CAS^n), consider the cascade subproblem $SCAS^n$:

$$(SCAS^n) \quad Z^n = \sum_{n' < n} \sum_{t \in TF^{n'}} h_t X_t^{n'} + \max \sum_{t \in TC^n} h_t X_t$$

$$s.t. \quad \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \leq s_t - \sum_{n' < n} \sum_{t' \in TS_t \cap TF^{n'}} a_{tt'} X_{t'}^{n'} \quad \forall t \in TC^n \qquad (SCAS^n.1)$$

$$X_t \geq 0 \qquad \forall t \in TC^n. \qquad (SCAS^n.2)$$

In addition to the formulation differences, $SCAS^n$ differs from CAS^n by a constant term in the objective function. When demand is reduced by previous subproblems, each CAS^n optimal objective value becomes progressively lower. The basic staircase model $SCAS^n$ on the other hand, must explicitly incorporate a contribution from previous subproblems. The solutions from these previous subproblems $n' < n$ are summed only over the set $TF^{n'}$ in order to avoid "double counting" columns indexed by periods inside the cascade overlap.

Let $X_t^S \ \forall t \in T$ and $X_t^n \ \forall t \in TC^n$ solve S and $SCAS^n$, respectively (as before, assume $SCAS^n$ has a finite optimal solution for all $n \in NC$). The following theorem shows that the proximal cascade solution value bounds the monolith solution value from below. Additionally, if $a_{tt'} \geq 0 \ \forall t, t'$, a proximal cascade also provides an upper bound to the monolith solution value. The upper bound is the sum of non-constant objective terms from all periods in all subproblems (thus it includes cascade overlap double counting). The lower bound is Z^N, the sum of objective terms from the non-overlapping periods in all subproblems.

Theorem 2.6

$$Z^S \geq Z^N, \quad \text{and if } a_{tt'} \geq 0 \; \forall t, t', \; \sum_{n \in NC} \sum_{t \in TC^n} h_t X_t^n \geq Z^S.$$

Proof: For the first inequality,

$$\begin{aligned}
Z^S &= \max \sum_{t \in T} h_t X_t \\
&\quad s.t. \;\; (S.1), (S.2) \\
&\geq \max \sum_{t \in T} h_t X_t \qquad (SP.1) \\
&\quad s.t. \;\; (S.1), (S.2) \\
&\quad X_t = X_t^n \;\; \forall n < N, t \in TF^n \\
&= \sum_{n' < N} \textstyle\sum_{t \in T} h_t X_t^{n'} + \max \sum_{t \in TC^N} h_t X_t \qquad (SP.2) \\
&\quad s.t. \;\; (SCAS^N.1), (SCAS^N.2) \\
&= Z^N.
\end{aligned}$$

SP.1 is a restriction of the original problem because all of the solution values are fixed except for the last subproblem's values. As with CAS^n, $SCAS^n$ is feasible to the monolith since the rows of the subproblems jointly enforce the rows S.

To show the second inequality, begin with the sum of the non-constant objective terms from the proximal cascade subproblems:

$$\sum_{n \in NC} \sum_{t \in TC^n} h_t X_t^n = \sum_{n \in NC} \max \left[\begin{array}{l} \sum_{t \in TC^n} h_t X_t \\ s.t. \;\; (SCAS^n.1), (SCAS^n.2) \end{array} \right] \qquad (SL.1)$$

$$\geq \sum_{n \in NC} \max \left[\begin{array}{l} \sum_{t \in TC^n} h_t X_t \\ s.t. \;\; (SCAS^n.1), (SCAS^n.2) \\ X_t = 0 \quad \forall t \in TC^n \cap TC^{n-1} \end{array} \right] \qquad (SL.2)$$

$$= \sum_{n \in NC} \max \left[\begin{array}{ll} \sum_{t \in TC^n \setminus TC^{n-1}} h_t X_t & \\ s.t. \sum_{t' \in TS_t \cap (TC^n \setminus TC^{n-1})} a_{tt'} X_{t'} \leq s_t & \forall t \in TC^n \setminus TC^{n-1} \\ X_t \geq 0 & t \in TC^n \setminus TC^{n-1} \end{array} \right] \qquad (SL.3)$$

$$\geq \sum_{n \in NC} \max \left[\begin{array}{ll} \sum\limits_{t \in TC^n \setminus TC^{n-1}} h_t X_t & \\ s.t. \sum\limits_{t' \in TS_t \cap TC^{n-1}} a_{tt'} X^S_{t'} & \\ + \sum\limits_{t' \in TS_t \cap (TC^n \setminus TC^{n-1})} a_{tt'} X_{t'} \leq s_t & \forall t \in TC^n \setminus TC^{n-1} \\ X_t = X^S_t & \forall t \in TC^n \cap TC^{n+1} \\ X_t \geq 0 & \forall t \in TC^n \setminus TC^{n-1} \end{array} \right] \quad (SL.4)$$

$$= \max \sum_{t \in T} h_t X_t$$
$$s.t. \sum_{t' \in TS} a_{tt'} X_{t'} \leq s_t \quad \forall t \in T \quad (SL.5)$$
$$X_t \geq 0 \qquad \forall t \in T$$

$$= Z^S.$$

SL.2 is a restriction of *SL.1* since all of the columns indexed by periods of the leading subproblem overlaps are set to 0. This is a nontrivial restriction if the cascade overlap is large. *SL.3* is a restatement of *SL.2*, since no resources are used by columns set to 0, which include any columns that might use resources in the remaining active periods. Additionally, the row domain of the staircase constraint from *SL.3* may now exclude the overlap rows, since they contain only constants. These are the only rows that could include fixed terms from previous subproblems, so that term may be dropped. *SL.4* further restricts the problem by including resource consumption of X^S_t from columns of the preceding subproblem's staircase overlap. *SL.4* also restricts the problem by fixing (to X^S_t) all the columns that appear in the *succeeding* subproblem's rows. Finally, *SL.5* reflects that a subproblem whose overlapping staircase values are set to the optimal solution (on either side) must produce optimal values when solved.

□

This result provides an optimistic bound (upper for a maximization problem) and a feasible bound (lower for a maximization) on the monolith, obtained for the computational cost of solving the proximal cascade. However, the usefulness of the optimistic bound is reduced by the fact that it tends to be tighter with a minimal cascade overlap ($v = m$), while the feasible bound tends to be tighter with a large cascade overlap. Additionally, the optimistic bound requires non-negativity of the technological coefficients, which also restricts

its applicability. The next section describes an optimistic bound with wide applicability—one that can be used on any staircase problem.

D. LAGRANGIAN CASCADE LOWER BOUND

1. Development

Lagrangian relaxation has long been used to bound linear and integer programs by solving partitioned subproblems. By partitioning subproblems along temporal lines, each can be solved separately by (Lagrangian) relaxing rows that link active periods from different subproblems. The structure of a staircase problem facilitates this, since most rows link only a few proximal time periods.

Multi-commodity elastic-demand staircase problems complicate relaxation along temporal lines because the demand rows link many time periods. However, the rows are elastic, which establishes bounds on the corresponding dual variables. Consequently, appropriate penalties in the relaxed problem stay within those limits.

The biggest advantage of using Lagrangian relaxation in a cascade is the availability of dual information from the associated proximal cascade. One of the weaknesses of Lagrangian relaxation is the computational effort required for the multiplier search. That search is circumvented by the availability of the proximal cascade's dual variables.

Consider once again problem B with finite optimal solution $X_{it}^B,\ P_i^B \quad \forall i,t:$

$$
\begin{aligned}
Z^B = \quad \min \quad & \sum_{i \in I} h_i P_i & & & & \\
s.t. \quad & \sum_{t \in T_i} X_{it} + P_i = d_i & \forall i \in I \quad & (\alpha_i) \quad & (B.1) \\
& \sum_{i \in I} \sum_{t' \in T_i \cap TS_t} a_{itt'} X_{it'} \leq s_t & \forall t \in T \quad & (\beta_t) \quad & (B.2) \\
& X_{it}, P_i \geq 0 \ \ \forall i \in I\ , t \in T. & & & (B.3)
\end{aligned}
$$

Dual variables are denoted $\alpha_i,\ \beta_t$.

Consider a partition of T into L subproblems (Figures 4 and 5):

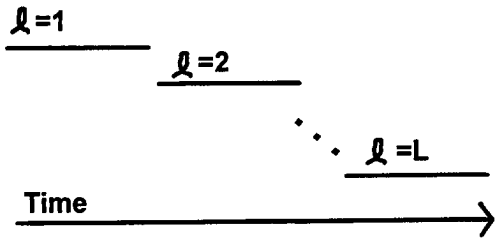

Figure 4. A Lagrangian cascade partitions the rows and columns of a monolith into many Lagrangian subproblems of contiguous time periods. Overlapping rows are Lagrange-relaxed.

The following notation and Figure 5 are also useful:

$firstl^{\ell}$	The first time period in subproblem ℓ
$lastl^{\ell}$	The last time period in subproblem ℓ
$lwid$	$\max_{\ell}\left[lastl^{\ell} - firstl^{\ell}\right] + 1$, the Lagrangian cascade subproblem width
TR^{ℓ}	$\{t : firstl^{\ell} \leq t < (firstl^{\ell} + m)\}, \ell \neq 1$. The Lagrange-relaxed set, the set of early periods of subproblem ℓ where staircase rows overlap subproblem $\ell - 1$
TL^{ℓ}	$\{t : \max(t \in TR^{\ell}) < t \leq lastl^{\ell}\}$ The enforced set, the set of later periods of ℓ where staircase rows do not overlap subproblem $\ell - 1$
TRL^{ℓ}	$TR^{\ell} \cup TL^{\ell}$ The active index set of subproblem ℓ
TO^{ℓ}	$\{t : firstl^{\ell} - m \leq t < firstl^{\ell}\}$ The extended set, the set of active periods in subproblem $\ell - 1$ where staircase rows of subproblem ℓ overlap
$TRL^{\ell} \cup TO^{\ell}$	The extended-active set
IL^{ℓ}	$\{i : T_i \cap TRL^{\ell} \neq \emptyset, T_i \cap TRL^{\ell+1} = \emptyset\}$ Partition of I into subproblems This scheme places i into the last subproblem in i's production window
CL	$\{1, ..., L\}$, The set of Lagrangian cascade subproblems

The active index sets (TRL^{ℓ}) may be chosen to closely correspond to the proximal cascade active sets, TC^{n}, or can be intentionally offset from those sets in order to improve the latter's dual variables. These strategies are discussed in the next chapter. Given these

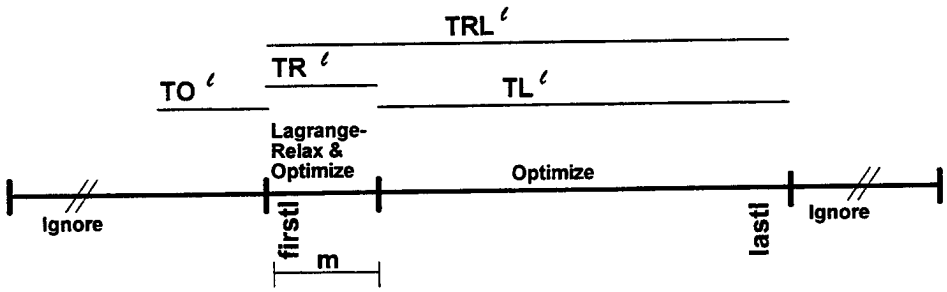

Figure 5. A single Lagrangian cascade subproblem includes columns indexed by $t \in TRL^{\ell}$ (the active set), and rows indexed by $t \in TL^{\ell}$. Other rows are relaxed, and a Lagrangian penalty is applied to the objective function coefficients of the associated active columns (referred to as Lagrange-relaxed rows). An extension tightens the Lagrangian cascade bound by activating "extended constraint" rows indexed by $t \in TR^{\ell}$ (the Lagrange-relaxed set). These rows use "duplicate" columns indexed by $t \in TO^{\ell}$ in order to preserve the relaxation.

sets, define the Lagrangian cascade subproblem LC:

$$
\begin{aligned}
Z^{LC} = \min \quad & \sum_{i \in I} h_i P_i && (LC.1) \\
& + \sum_{i \in I} \alpha_i \left(d_i - \sum_{t \in T_i} X_{it} - P_i \right) \\
& + \sum_{t \in (\cup_\ell TR^\ell)} \beta_t \left(s_t - \sum_{i \in I} \sum_{t' \in T_i \cap TS_t} a_{itt'} X_{it'} \right) \\
s.t. \quad & \sum_{i \in I} \sum_{t' \in T_i \cap TS_t} a_{itt'} X_{it'} \leq s_t && \forall t \in \cup_\ell TL^\ell \quad (LC.2) \\
& X_{it}, P_i \geq 0 \ \ \forall i \in I \ , t \in T. && (LC.3)
\end{aligned}
$$

The objective $(LC.1)$ seeks to minimize the sum of unsatisfied demand, plus Lagrangian penalties either associated with demand rows, or the staircase rows of the relaxed set. The remaining structural constraints include only the staircase rows of the enforced set.

Because all of the linking rows between subproblems are Lagrange-relaxed, LC decomposes into L subproblems with $Z^{LC} = \sum_\ell Z^\ell$. Note α_i is bounded above by h_i, and thus the P_i term is not favorable, and will remain at 0. It is left in the formulation for completeness. The subproblems LC^{ℓ} are defined as

$$
\begin{aligned}
Z^{\ell} = \min \quad & \sum_{i \in IL^{\ell}} P_i(h_i - \alpha_i) + \sum_{i \in IL^{\ell}} \alpha_i d_i && (LC^{\ell}.1)\\
& - \sum_{i \in I} \sum_{t \in TRL^{\ell}} \alpha_i X_{it} + \sum_{t \in TR^{\ell}} \beta_t s_t \\
& - \sum_{i \in I} \sum_{t \in TR^{\ell} \cup TR^{\ell+1}} \sum_{t' \in T_i \cap TS_t \cap TRL^{\ell}} \beta_t a_{itt'} X_{it'} \\
s.t. \quad & \sum_{i \in I} \sum_{t' \in T_i \cap TS_t} a_{itt'} X_{it'} \leq s_t && \forall t \in TL^{\ell} \quad (LC^{\ell}.2)\\
& X_{it}, P_i \geq 0 \ \ \forall i \in IL^{\ell}, t \in TRL^{\ell}. && (LC^{\ell}.3)
\end{aligned}
$$

Relaxing the problem in this manner allows the tractable computation of a lower bound on Z^B. By the theorem of weak Lagrangian duality [Parker and Rardin, 1988, p. 206],

$$Z^{LC} = \sum_{\ell} Z^{\ell} \leq Z^B.$$

A Lagrangian cascade proceeds as follows (a detailed pseudocode is given in Chapter III):

For each $\ell \in CL$ {
  Define and solve subproblem LC^{ℓ} given above
  Record the value of Z^{ℓ}
}
Output the Lagrangian cascade solution value: $\sum_{\ell} Z^{\ell}$.

As stated earlier, the quality of this bound depends in large measure on the quality of the dual variables. These variables, in turn, depend on the quality of the proximal cascade solution. As the proximal cascade solution tends toward the optimal monolith solution, the associated duals will tend toward the optimal monolith dual solution. Hence, there is strong incentive for making the proximal cascade solution as close to the monolith solution as possible.

2. Improving the Lagrangian Cascade Bound

Although optimal Lagrange multipliers ensure a tight lower bound on the problem under consideration, small deviations in multiplier accuracy may produce an unacceptable, or even unbounded result. This may be avoided by bounding the feasible region of the Lagrangian cascade subproblems. This section addresses two bounding techniques, *extended constraints* and *demand bounding*.

a. Extended Constraints

A Lagrangian relaxation cannot be unbounded if all its variables are bounded. Using a Lagrangian cascade, we show here that a simple and effective bound on variables is generated by extending the staircase constraint enforcement into each subproblem's Lagrange-relaxed set, TR^{ℓ}. However, to avoid a problem restriction, associated columns indexed by periods of the overlap set (TO^{ℓ}) are not identical to their monolith counterparts. These columns are "duplicates," and are denoted $\widetilde{X}_{it}$. This method of generating duplicate columns for the purpose of bounding variables inside the active index set is described below.

Consider problem $\widetilde{B}$, which is identical to B, but with constraint blocks $\widetilde{B}.4$ and $\widetilde{B}.5$ added:

$$
\begin{aligned}
Z^{\tilde{B}} = \\
\min \quad & \sum_{i \in I} h_i P_i \\
s.t. \quad & \sum_{t \in T_i} X_{it} + P_i = d_i && \forall i \in I && (B.1) \\
& \sum_{i \in I} \sum_{t' \in T_i \cap TS_t} a_{itt'} X_{it'} \leq s_t && \forall t \in T && (B.2) \\
& X_{it}, P_i \geq 0 \ \ \forall i \in I \ , t \in T && && (B.3) \\
& \sum_{i \in I} \sum_{t' \in T_i \cap TS_t \cap TO^{\ell}} a_{itt'} \widetilde{X}_{it'} + \sum_{i \in I} \sum_{t' \in T_i \cap TS_t \cap TR^{\ell}} a_{itt'} X_{it'} \leq s_t && \forall t \in \textstyle\bigcup_{\ell} TR^{\ell} && (\widetilde{B}.4) \\
& \widetilde{X}_{it} \geq 0 \ \ \forall \ell \in CL, \ i \in I \ , t \in T \textstyle\bigcup_{\ell} TR^{\ell}. && && (\widetilde{B}.5)
\end{aligned}
$$

$\widetilde{B}.4$ duplicates all staircase rows for all of the relaxed sets $\bigcup_{\ell} TR^{\ell}$. However, within each subproblem, only the columns indexed by $t \in TRL^{\ell}$ correspond to monolith columns. Columns indexed by $t \in TO^{\ell}$ are duplicated; duplicates do not appear in other subproblems. Because of duplication, these rows cannot restrict the solution to B.

Theorem 2.7

$$Z^{B} \geq Z^{\tilde{B}}.$$

Proof: Let $\widetilde{X}_{it} = X^{B}_{it} \quad \forall i, t \in \bigcup_{\ell} TR^{\ell}$. Since X^{B}_{it} satisfies $B.2$ from the original problem, $\widetilde{X}_{it}$ must satisfy $(B.4)$, which means X^{B}_{it} is feasible to $\widetilde{B}$. Thus, $Z^{\tilde{B}}$ can be no worse than Z^{B}. □

In fact, $Z^B = Z^{\tilde{B}}$, since the duplicate columns do not contribute to the objective, nor do they allow the original columns to further contribute to the objective. However, this is not central to the overall result, which is to show that a Lagrangian relaxation of $\widetilde{B}$ is still a relaxation of B.

Since any Lagrangian relaxation of $\widetilde{B}$ provides a lower bound for $\widetilde{B}$, one that relaxes $B.1$ $\forall i$, and $B.2$ $\forall t \in \bigcup_\ell TR_\ell$ provides such a bound. Defining the solution value of this relaxation to be $\widetilde{Z}^{LC}$, we have

$$Z^B \geq Z^{\tilde{B}} \geq \widetilde{Z}^{LC}.$$

The solution to the relaxation of $\widetilde{B}$ offers the benefit of bounding all variables in the Lagrangian cascade. Since its implementation only involves generating duplicate variables as the Lagrangian cascade progresses, this strategy may provide significant benefit with minimal computational effort.

The extended constraints result has wide applicability to Lagrangian relaxation. Consider the following staircase problem:

$$\begin{array}{rcrrrrrl} Z^* & = & \max & c_1^T x_1 & +c_2^T x_2 & +c_3^T x_3 & & \\ & & s.t. & A_{11}x_1 & & & \leq & b_1 \\ & & & A_{21}x_1 & +A_{22}x_2 & & \leq & b_2 \\ & & & & A_{32}x_2 & +A_{33}x_3 & \leq & b_3 \\ & & & x_1, & x_2, & x_3 & \geq & 0. \end{array}$$

This problem can be separated into two subproblems by relaxing the second constraint:

$$\begin{array}{rcrrrrrl} Z^{LR} & = & \max & c_1^T x_1 & +c_2^T x_2 & +c_3^T x_3 & + & \beta_2 (b_2 - A_{21}x_1 - A_{22}x_2) \\ & & s.t. & A_{11}x_1 & & & \leq & b_1 \\ & & & & A_{32}x_2 & +A_{33}x_3 & \leq & b_3 \\ & & & x_1, & x_2, & x_3 & \geq & 0. \end{array}$$

Alternatively, we can form separable subproblems by duplicating x_1 and using the extended

constraints Lagrangian relaxation:

$$\begin{array}{rcccccccl} Z^{ELR} & = & \max & c_1^T x_1 & & +c_2^T x_2 & +c_3^T x_3 & + & \beta_2 \left(b_2 - A_{21}x_1 - A_{22}x_2\right) \\ & & s.t. & A_{11}x_1 & & & & \leq & b_1 \\ & & & & A_{21}\widetilde{x}_1 & +A_{22}x_2 & & \leq & b_2 \\ & & & & & A_{32}x_2 & +A_{33}x_3 & \leq & b_3 \\ & & & x_1, & \widetilde{x}_1, & x_2, & x_3 & \geq & 0. \end{array}$$

Both Z^{LR}, and Z^{ELR} provide upper bounds for Z^*, but Z^{ELR} provides a tighter bound by Theorem II.7:

$$Z^* \leq Z^{ELR} \leq Z^{LR}.$$

To illustrate the benefits of extended constraints, consider the following staircase LP:

$$\begin{array}{rccccc} \max & 2X_1 & +4X_2 & +X_3 & & \\ s.t. & X_1 & & & \leq & 2 \\ & X_1 & +X_2 & & \leq & 3 \quad (\beta_2) \\ & & X_2 & +X_3 & \leq & 4 \\ & X_1, & X_2, & X_3 & \geq & 0. \end{array}$$

A solution to this problem is: $X_2^* = 3,\ X_3^* = 1$, with value 13. Lagrangian relaxation of the second row results in the following for $\beta_2 \geq 0$:

$$\begin{array}{rccccc} \max & 2X_1 & +4X_2 & +X_3 & + & \beta_2(3 - X_1 - X_2) \\ s.t. & X_1 & & & \leq & 2 \\ & & X_2 & +X_3 & \leq & 4 \\ & X_1, & X_2, & X_3 & \geq & 0. \end{array}$$

When $\beta_2 = 1$, the above may be rewritten as

$$\begin{array}{cccccccccc} 3 & + \max & X_1 & & & & + \max & 3X_2 & +X_3 & \\ & s.t. & X_1 & \leq & 2 & & s.t. & X_2 & +X_3 & \leq & 4 \\ & & X_1 & \geq & 0 & & & X_2, & X_3 & \geq & 0. \end{array}$$

This has a solution $\overline{X}_1 = 2,\ \overline{X}_2 = 4$, with value 17, which is an optimistic bound on the first problem, $Z^* = 13$. However, the bound may be tightened by duplicating X_1 with $\widetilde{X}_1$,

and incorporating the method of extended constraints:

$$\begin{array}{rlll} 3 + \max & X_1 & + \max & 3X_2 + X_3 \\ s.t. & X_1 \leq 2 & s.t. & \widetilde{X}_1 + X_2 \leq 3 \\ & X_1 \geq 0 & & X_2 + X_3 \leq 4 \\ & & & \widetilde{X}_1 \quad X_2, \quad X_3 \geq 0. \end{array}$$

This has solution $X_1 = 2,\ X_2 = 3$, with value 14, resulting in a tighter bound than 17.

b. Demand Bounding

In addition to the method of extended constraints described above, the quality of the Lagrangian cascade solution may be improved by assuring that each subproblem satisfies no more demand than the total required by a Lagrange-relaxed demand row.

Consider problem $\overline{B}$, which is identical to B, but with additional constraint stipulated by $\overline{B}.4$:

$$\begin{array}{lll} (\overline{B}) \quad Z^{\bar{B}} = \min & \sum_{i \in I} h_i P_i & \\ s.t. & \sum_{t \in T_i} X_{it} + P_i = d_i \quad \forall i \in I & (B.1) \\ & \sum_{t \in T_i \cap TRL_\ell} X_{it} \leq d_i \quad \forall i \in I, \forall \ell \in CL & (\overline{B}.4) \\ & (B.2), (B.3). & \end{array}$$

Note that $\overline{B}.4$ is redundant given $B.1$ in this formulation, but ceases to be so when $B.1$ is relaxed. Hence, B and $\overline{B}$ are equivalent, but the relaxation of $\overline{B}$ provides a better lower bound on Z^B.

Demand bounding also applies to Lagrangian relaxations of more general mathematical programs. Given that all elements of A_{21} and A_{22} are non-negative, consider how demand bounding can tighten the solution bound of a simple staircase problem:

$$\begin{array}{rlllll} Z^* = \max & c_1^T x_1 & + c_2^T x_2 & + c_3^T x_3 & & \\ s.t. & A_{11} x_1 & & & \leq & b_1 \\ & A_{21} x_1 & + A_{22} x_2 & & \leq & b_2 \\ & & A_{32} x_2 & + A_{33} x_3 & \leq & b_3 \\ & x_1, & x_2, & x_3 & \geq & 0 \end{array}$$

$$
\begin{array}{rlrrrrl}
\leq & \max & c_1^T x_1 & +c_2^T x_2 & +c_3^T x_3 & + & \beta_2 \left(b_2 - A_{21}x_1 - A_{22}x_2\right) \\
& s.t. & A_{11}x_1 & & & \leq & b_1 \\
& & A_{21}x_1 & & & \leq & b_2 \\
& & & A_{22}x_2 & & \leq & b_2 \\
& & & A_{32}x_2 & +A_{33}x_3 & \leq & b_3 \\
& & x_1, & x_2, & x_3 & \geq & 0
\end{array}
$$

$$
\begin{array}{rlrrrrl}
\leq & \max & c_1^T x_1 & +c_2^T x_2 & +c_3^T x_3 & + & \beta_2 \left(b_2 - A_{21}x_1 - A_{22}x_2\right) \\
& s.t. & A_{11}x_1 & & & \leq & b_1 \\
& & & A_{32}x_2 & +A_{33}x_3 & \leq & b_3 \\
& & x_1, & x_2, & x_3 & \geq & 0.
\end{array}
$$

Z^* is bounded above by the Lagrangian relaxation with demand bounding (middle), which is bounded above by the un-enhanced Lagrangian relaxation (bottom).

E. SUMMARY

This chapter develops bounds for staircase problems using proximal and Lagrangian cascades. A proximal cascade provides an upper bound (if a minimization problem) by solving a sequence of subproblems. Under the restricted condition of non-negativity of the constraint coefficients, modifying the proximal cascade solution value also provides a lower bound on the monolith solution value.

A Lagrangian cascade provides an optimistic bound for the staircase problems described in this chapter. Lagrangian cascades avoid traditional multiplier searches by using dual information from the proximal cascade. Together, proximal and Lagrangian cascades offer an alternative to solving a linear program monolith.

III. PROXIMAL AND LAGRANGIAN CASCADE HEURISTICS

This chapter serves as a transition between theory and implementation of the proximal and Lagrangian cascades. Each cascade type is dealt with separately, beginning with a description and pseudocode. A discussion of cascade parameter selection ensues, followed by an overview of model characteristics that may allow cascade solutions of good quality.

We use problem B from Chapter II as an example to develop the heuristics. Time serves as the cascade set for this multi-commodity elastic-demand staircase problem. Demand rows (indexed by i) have the null cascade index. Additionally, coefficients ($a_{itt'}$) may be negative and rows may represent equalities or inequalities. We assume that staircase rows are indexed by the greatest time period of any associated column.

A. THE PROXIMAL CASCADE

1. Description

We initialize a proximal cascade with the selection of two parameters: 1) cascade width, *caswid,* and 2) cascade overlap, v. For subproblem n, define

$$\begin{aligned} \mathit{firstp}^n &= (n-1)\cdot(\mathit{caswid} - v) + 1 \\ \mathit{lastp}^n &= \min\left[T, (n-1)\cdot(\mathit{caswid} - v) + \mathit{caswid}\right] \end{aligned}$$

for $n = \{1, ..., N\}$ such that $\mathit{lastp}^{N-1} < T$, $\mathit{lastp}^N = T$. Using these definitions, the sets TC^n and TF^n are as defined in Chapter II, Section B.2.

A proximal cascade subproblem consists of active columns and active rows that have been adjusted for the levels of fixed columns. Fixing a column implies adjusting the right-hand-sides of all associated active rows for resources consumed, and adjusting the objective function value by the objective contribution of the column level.

The following rules provide a guide for solving a proximal cascade:

- Form the first subproblem:

  Activate all columns indexed by $t \in TC^1$ (production columns, X_{it}), plus all columns indexed only by i where $TC^1 \cap T_i \neq \emptyset$ (elastic columns, P_i).

  Activate rows indexed by $t \in TC^1$ (staircase rows), or $i \ni TC^1 \cap T_i \neq \emptyset$ (demand rows). In Chapter II's description of proximal cascades, demand rows and elastic columns are active in every subproblem; selectively activating these rows and columns reduces subproblem size without altering the cascade solution.

  Solve the subproblem.

- Update the cascade to subproblem n:

  Relax active rows indexed by: 1) $t \notin TC^n$, or 2) $i \ni TC^n \cap T_i = \emptyset$. Fix active columns that meet either of these two criteria.

  Activate rows indexed by $t \in TC^n$ or $i \ni TC^n \cap T_i \neq \emptyset$. Activate columns that meet the same criteria.

  Solve the subproblem.

  Repeat the cascade update until the final subproblem is solved.

The objective value of the final subproblem is the proximal cascade solution value. A feasible proximal cascade solution is also feasible to the monolith because all columns are fixed only after satisfying associated rows.

2. Pseudocode for a Proximal Cascade

We supplement the guide above with pseudocode for a proximal cascade. This code makes use of the same notation and assumptions as before. Additionally, define a *candidate* row or column as one that has never been active.

Procedure **Proximal Cascade**
INPUT: *caswid*, v, monolith LP B
OUTPUT: proximal cascade objective value, proximal cascade solution
{
$upper\_bound = 0$
$lastp = 0$
$n = 1$
while $(lastp < T)$ {
  $firstp = (n-1) \cdot (caswid - v) + 1$
  $lastp = \min[T, (n-1) \cdot (caswid - v) + caswid]$
  $TC = \{t : firstp \leq t \leq lastp\}$
  if $(n > 1)$ {
    for each active **row**, {
      if ((**row** indexed by $t \in TC$) or (**row** indexed by $i \ni TC \cap T_i = \emptyset$)) {

```
            relax row
          }
        }
        for each active column {
          if ((column indexed by t ∈ TC) or (column indexed by i ∋ TC ∩ T_i = ∅)) {
            add column level · column objective coefficient to upper_bound
            for each active row associated with column {
              adjust row RHS by subtracting column level · column coefficient
            }
            record column level
            make column inactive
          }
        }
      }
      for each candidate row {
        if ((row indexed by t ∈ TC^n) or (row indexed by i ∋ TC ∩ T_i ≠ ∅)) {
          make row active
        }
      }
      for each candidate column {
        if ((column indexed by t ∈ TC) or (column indexed by i ∋ TC ∩ T_i ≠ ∅)) {
          make column active
        }
      }
      solve subproblem
      n = n + 1
    }
    add final subproblem's active objective terms to upper_bound
    record final subproblem's active column levels
    report upper_bound as the proximal cascade objective value
    report recorded column levels as the proximal cascade solution
    }
```

3. Parameter Selection

a. Selection of Cascade Width, *caswid*

Two considerations often limit cascade width, *caswid* . The appropriate level of model myopia dictates a corresponding cascade width. Computational considerations can also limit cascade width; a few large subproblems take longer to solve than many short ones

if the cascade overlap is small.

Usually, the proximal cascade solution quality increases as cascade width increases. Intuitively this is suggested by the fact that selecting *caswid* $= T$ results in solving the monolith. However, smaller cascade widths can (counter-intuitively) improve the proximal cascade solution quality. Consider a 2-commodity, 4-period instance of problem B:

$$
\begin{array}{llllllllll}
Z^B = & \min & & & & & & P_X & +P_Y & \\
s.t. & & & & & & & & & \\
X_1 & +X_2 & & +X_3 & & +X_4 & & +P_X & & = 10 \\
 & Y_1 & & +Y_2 & & +Y_3 & +Y_4 & & +P_Y & = 20 \\
2X_1 & +4Y_1 & & & & & & & & \leq 20 \\
2X_1 & +4Y_1 & +10X_2 & +10Y_2 & & & & & & \leq 20 \\
 & & 10X_2 & +10Y_2 & +2X_3 & +10Y_3 & & & & \leq 20 \\
 & & & & 2X_3 & +10Y_3 & +10X_4 & +Y_4 & & \leq 20 \\
X_1, & Y_1, & X_2, & Y_2 & X_3, & Y_3, & X_4, & Y_4, & P_X, \quad P_Y & \geq 0.
\end{array}
$$

Here, $X_1^B = 10,\ Y_4^B = 20,\ Z^B = 0$ solves the above (unstated variable levels are 0 both here and below).

Now consider a proximal cascade solution with *caswid* $= 2,\ v = 1$ (3 subproblems):

$$
\begin{array}{llllllll}
Z^{B1} = & \min & & & & P_X & +P_Y & \\
 & s.t. & X_1 & & +X_2 & +P_X & & = 10 \\
 & & & Y_1 & +Y_2 & & +P_Y & = 20 \\
 & & 2X_1 & +4Y_1 & & & & \leq 20 \\
 & & 2X_1 & +4Y_1 & +10X_2 \quad +10Y_2 & & & \leq 20 \\
 & & X_1, & Y_1, & X_2, \quad Y_2 & P_X, & P_Y & \geq 0
\end{array}
$$

(note: $X_1^{B1} = 10,\ P_Y^{B1} = 20,\ Z^{B1} = 20$),

$$
\begin{array}{llllll}
Z^{B2} & = \min & & P_X & +P_Y & \\
s.t. & & & & & \\
X_2 & & +X_3 & +P_X & & = 10 - X_1^{B1} \\
 & +Y_2 & & & +P_Y & = 20 - Y_1^{B1} \\
 & 10X_2 \quad +10Y_2 & & & & \leq 20 - 2X_1^{B1} - 4Y_1^{B1} \\
 & 10X_2 \quad +10Y_2 & +2X_3 \quad +10Y_3 & & & \leq 20 \\
 & X_2, \quad Y_2, & X_3, \quad Y_3, & P_X, & P_Y & \geq 0
\end{array}
$$

(note: $X_1^{B1} = 10,\ Y_3^{B2} = 2,\ P_Y^{B2} = 18,\ Z^{B2} = 18$), and

$$
\begin{array}{llllllll}
Z^{B3} = & \min & & & P_X & +P_Y & & \\
s.t. & & & & & & & \\
X_3 & & +X_4 & & +P_X & & = 10 - X_1^{B1} - X_2^{B2} & \\
 & +Y_3 & & +Y_4 & & +P_Y & = 20 - Y_1^{B1} - Y_2^{B2} & \\
2X_3 & +10Y_3 & & & & & \leq 20 - 10X_2^{B2} - 10Y_2^{B2} & \\
2X_3 & +10Y_3 & +10X_4 & +Y_4 & & & \leq 20 & \\
X_3, & Y_3, & X_4, & Y_4, & P_X, & P_Y & \geq 0 &
\end{array}
$$

(note: $Y_4^{B3} = 20,\ Z^{B3} = 0$). Thus, the proximal cascade solution is $X_1^{B1} = 10,\ Y_4^{B3} = 20,\ Z^{B3} = 0$, as in the monolith.

However, setting *caswid* $= 3$ (and $v = 1$, resulting in 2 subproblems) produces a larger objective value:

$$
\begin{array}{llllllllll}
Z^{B1'} = & \min & & & & & & P_X & +P_Y & \\
s.t. & X_1 & +X_2 & & +X_3 & & & +P_X & & = 10 \\
 & Y_1 & & +Y_2 & & +Y_3 & & & +P_Y & = 20 \\
 & 2X_1 & +4Y_1 & & & & & & & \leq 20 \\
 & 2X_1 & +4Y_1 & +10X_2 & +10Y_2 & & & & & \leq 20 \\
 & & & 10X_2 & +10Y_2 & +2X_3 & +10Y_3 & & & \leq 20 \\
 & X_1, & Y_1, & X_2, & Y_2 & X_3, & Y_3, & P_X, & P_Y & \geq 0
\end{array}
$$

(note: $Y_1^{B1'} = 5,\ X_3^{B1'} = 10,\ P_Y^{B1'} = 15,\ Z^{B1'} = 15$), and

$$
\begin{array}{llllllll}
Z^{B2'} = & \min & & & P_X & +P_Y & & \\
s.t. & & & & & & & \\
X_3 & & +X_4 & & +P_X & & = 10 - X_1^{B1'} - X_2^{B1'} & \\
 & +Y_3 & & +Y_4 & & +P_Y & = 20 - Y_1^{B1'} - Y_2^{B1'} & \\
2X_3 & +10Y_3 & & & & & \leq 20 - 10X_2^{B1'} - 10Y_2^{B1'} & \\
2X_3 & +10Y_3 & +10X_4 & +Y_4 & & & \leq 20 & \\
X_3, & Y_3, & X_4, & Y_4, & P_X, & P_Y & \geq 0 &
\end{array}
$$

(note: $X_3^{B2'} = 2.5,\ Y_4^{B2'} = 15,\ P_X^{B2'} = 7.5,\ Z^{B2'} = 7.5$). Consequently, this proximal cascade solution is $Y_1^{B1'} = 5,\ X_3^{B2'} = 2.5, Y_4^{B2'} = 15, P_X^{B2'} = 7.5, Z^{B2'} = 7.5$. Note that the second subproblem is shortened by the problem's last period, T.

Analysis of the above results shows that the *caswid* $= 3$ case is "tricked" into producing commodity Y early in period 1, while the *caswid* $= 2$ case avoids this mistake. Thus, the more myopic cascade has higher solution quality than the less myopic one, in this instance.

b. Selection of Cascade Overlap, v

As with the cascade width, selection of the overlap parameter v affects the quality of the proximal cascade solution significantly. At one extreme, setting v =*caswid* -1 tends to produce higher quality solutions, because each subproblem moves forward only one time period, and re-solves columns of *caswid* -1 periods. This increases the ability of each subproblem to respond to new choices and restrictions posed by the added columns and rows.

Large cascade overlaps also preserve more of the optimal basis from subproblem to subproblem, so each solve may require fewer pivots if a simplex algorithm is used. However, even an advanced basis may not overcome the additional computations associated with large overlaps, so the overall solution time may be longer. Indeed, the results of the case study confirm this.

At the other extreme, the cascade overlap v may be set equal to the staircase overlap m. This approach minimizes the number of subproblems, but may lower solution quality. However, if all non-elastic rows have sense "$\leq$" and positive coefficients, feasibility is ensured by setting v to any value greater than or equal to m. This is shown by noting that any new staircase row (not previously active) of a subproblem includes columns from at most m periods prior to the staircase index t. By setting $v \geq m$, none of these columns are fixed; hence, all may be set to 0, satisfying the row trivially. On the other hand, if $v < m$, infeasibility may result.

Consider the following single commodity elastic-demand staircase problem with 5 periods and $m = 2$:

$$
\begin{array}{llccccccc}
Z^* = & \min & & & & & & P & & \\
 & s.t. & X_1 & +2X_2 & +X_3 & +X_4 & +X_5 & +P & = & 10 \\
 & & X_1 & & & & & & \leq & 2 \\
 & & X_1 & +X_2 & & & & & \leq & 2 \\
 & & X_1 & +X_2 & +X_3 & & & & \leq & 2 \\
 & & & +2X_2 & +X_3 & +X_4 & & & \leq & 2 \\
 & & & & +X_3 & +X_4 & +X_5 & & \leq & 2 \\
 & & X_1, & X_2, & X_3, & X_4, & X_5, & P & \geq & 0.
\end{array}
$$

Here, $X_1^* = 1,\ X_2^* = 1,\ X_5^* = 2,\ P^* = Z^* = 5.$

Now, consider a proximal cascade with *caswid* $= 3,\ v = 1$:

$$\begin{array}{lllllll} Z^1 = & \min & & & & P & \\ & s.t. & X_1 & +2X_2 & +X_3 & +P & = & 10 \\ & & X_1 & & & & \leq & 2 \\ & & X_1 & +X_2 & & & \leq & 2 \\ & & X_1 & +X_2 & +X_3 & & \leq & 2 \\ & & X_1, & X_2, & X_3, & P & \geq & 0 \end{array}$$

(here, $X_2^1 = 2,\ P^1 = Z^1 = 6$), and

$$\begin{array}{lllllll} Z^2 = & \min & & & P & & \\ & s.t. & X_3 & +X_4 & +X_5 & +P & = & 10 - X_1^1 - 2X_2^1 \\ & & X_3 & & & & \leq & 2 - X_1^1 - X_2^1 \\ & & X_3 & +X_4 & & & \leq & 2 - 2X_2^1 \\ & & X_3 & +X_4 & +X_5 & & \leq & 2 \\ & & X_3, & X_4, & X_5, & P & \geq & 0. \end{array} \qquad \text{(infeasible)}$$

This subproblem is infeasible since the right-hand-side of the third row is negative. The example shows that setting the cascade overlap less than the staircase overlap can, in some cases, result in infeasibility.

4. Desirable Model Characteristics for the Proximal Cascade

There are several model characteristics of the multi-commodity elastic-demand staircase problem that significantly affect the quality of the proximal cascade solution, the foremost being linkage. Demand and staircase rows with large widths link many time periods, requiring more rows to be active in multiple subproblems. Since later subproblems do not communicate these rows' resource costs to earlier subproblems, the earlier subproblem is more apt to make decisions that degrade solution quality.

Large staircase overlaps tend to reduce solution quality. Large overlaps result in more relaxed rows associated with columns of the active index set (at the end of a subproblem). Additionally, large overlaps cause more fixed columns from earlier subproblems

to alter active rows. Either condition increases the opportunity for earlier subproblems to make decisions that severely affect solutions of subsequent subproblems.

Cascade solution quality can be degraded when the model is formulated without time-discounted demand penalties. Consider a 2 commodity, 3 time period instance of B:

$$
\begin{array}{rrrrrrrrl}
Z^* = \min & & & & & P_1 & +P_2 & & \\
s.t. & X_{11} & & & & +P_1 & & = & 2 \\
 & & X_{22} & +X_{23} & & & +P_2 & = & 2 \\
 & X_{11} & & & & & & \leq & 2 \\
 & X_{11} & +X_{22} & & & & & \leq & 2 \\
 & & X_{22} & +X_{23} & & & & \leq & 2 \\
 & X_{11}, & X_{22}, & X_{23}, & P_1, & & P_2 & \geq & 0.
\end{array}
$$

This problem has solution $Z^* = 0,\ X^*_{11} = 2,\ X^*_{23} = 2$. Now define proximal cascade subproblems by letting *caswid* $= 2$, and $v = 1$:

$$
\begin{array}{rrrrrrl}
Z^1 = \min & & & P_1 & +P_2 & & \\
s.t. & X_{11} & & +P_1 & & = & 2 \\
 & & X_{22} & & +P_2 & = & 2 \\
 & X_{11} & & & & \leq & 2 \\
 & X_{11} & +X_{22} & & & \leq & 2 \\
 & X_{11}, & X_{22}, & P_1, & P_2 & \geq & 0
\end{array}
$$

One of the alternate optimal solutions to this subproblem is $Z^1 = 2,\ X^1_{22} = 2,\ P^1_1 = 2$. Another optimal solution is $Z^1 = 2,\ X^1_{11} = 2,\ P^1_2 = 2$.

$$
\begin{array}{rrrrrrl}
Z^2 = \min & & & P_1 & +P_2 & & \\
s.t. & & & P_1 & & = & 2 - X^1_{11} \\
 & X_{22} & +X_{23} & & +P_2 & = & 2 \\
 & X_{22} & & & & \leq & 2 - X^1_{11} \\
 & X_{22} & +X_{23} & & & \leq & 2 \\
 & X_{22}, & X_{23}, & P_1, & P_2 & \geq & 0.
\end{array}
$$

Subproblem 2's solution is monolith-optimal when $X^1_{11} = 2$, but not when $X^1_{11} = 0$. Ensuring that subproblem 1 chooses $X_{11} = 2$ can be accomplished by time-discounting the penalties. Changing the monolith's objective coefficient on P_2 to a value strictly between 0 and 1 would result in the correct prioritization of demand satisfaction by subproblem 1.

Finally, side constraints can reduce solution quality of a multi-commodity, elastic-demand staircase model that is solved by cascade. Rows associated with columns indexed by a single period present no difficulty; they are analogous to staircase rows with no over-

lap. On the other hand, rows that link many periods present a greater challenge, and must be assessed individually with respect to cascade feasibility. The case study of the next chapter exhibits an example of this challenge. The Air Force model includes a "utilization rate constraint," which, in the terminology of this chapter's example, limits the average utilization of production resources over many time periods. It is dealt with by aligning the periods over which utilization is averaged with the cascade subproblems. Chapter V also addresses these situations.

B. THE LAGRANGIAN CASCADE

We now describe how to select and solve Lagrangian cascade subproblems from the monolith. Unlike a proximal cascade subproblem, each Lagrangian cascade subproblem preserves none of the previous subproblem's solution; it activates an entirely new set of rows and columns. However, a Lagrangian cascade is complicated by objective function coefficient adjustments, demand bounding, and extended constraints.

1. Description

We initialize a Lagrangian cascade by specifying the Lagrangian cascade width, $lwid$. For subproblem ℓ, define

$$\begin{aligned} firstl^{\ell} &= (\ell - 1) \cdot lwid + 1 \\ lastl^{\ell} &= \min\left[T, \ell \cdot lwid\right] \end{aligned}$$

for $\ell = \{1, ..., L\}$ such that $lastl^{L-1} < T,\ lastl^{L} = T$. The sets TRL^{ℓ} and TO^{ℓ} are as defined in Chapter II, Section D.1.

Lagrangian cascade subproblems may be solved in any order. The following rules provide a guide to solving Lagrangian cascade subproblem ℓ:

- Activate rows whose associated columns are indexed by i, or by $t \in TRL$.
- Lagrange-relax rows that are associated with a column indexed by $t \in TRL^{\ell}$ and another column indexed by $t \notin TRL^{\ell}$.
- Activate *and* Lagrange-relax rows that serve as demand bounding rows or extended constraint rows (these techniques are discussed in Chapter II). Rows in-

dexed by i are demand rows; activate (and Lagrange-relax) these rows with sense "$\leq$" if some, but not all associated columns are indexed by $t \in TRL$. Activating extended constraint rows is more complex; these rows are activated if they have associated columns that meet criterion 3, described below.

- Activate columns that meet any of these three criteria:

  1) the column is indexed by $i \ni T_i \subseteq TRL^{\ell}$. These columns correspond to elastic penalty variables for demands that can only be met in this subproblem.

  2) the column is indexed by $t \in TRL^{\ell}$. These columns correspond to those of the active index set.

  3) the column is indexed by $t \in TO^{\ell}$. These correspond to duplicate columns of the "extended constraint" rows. Activate these columns with an objective function coefficient of 0. Activate rows associated with these columns if they are 1) indexed by $t \in TRL^{\ell}$, or 2) the corresponding sense is "$\leq$", and all coefficients are non-negative. These rules preserve the Lagrangian bound. For example, including an "equality" row indexed by $t \in TO^{\ell}$ might cause an infeasibility, since not all columns associated with this row are active.

- Solve the subproblem

The lower bound of a multiple-commodity elastic-demand staircase problem equals the sum of all Lagrangian cascade subproblem objective function values, plus all Lagrange-relaxed right-hand-sides multiplied by the associated Lagrange multipliers. The quality of that bound is dependent on *lwid*, the proximity of the multipliers used to the monolith-optimal multipliers, and the structure of the problem.

2. Pseudocode for a Lagrangian Cascade

We supplement the guide above with pseudocode for a Lagrangian cascade. This code makes use of the same notation and assumptions as the proximal cascade pseudocode

Procedure **Lagrangian Cascade**
INPUT: *lwid*, monolith LP B, Lagrange multipliers from the proximal cascade
OUTPUT: Lagrangian cascade objective value
{
$lower\_bound = 0$
$lastl = 0$
$\ell = 1$
while $(lastl < T)$ {
  $firstl = (\ell - 1) \cdot lwid + 1$
  $lastl = \min[T, \ell \cdot lwid]$
  $TRL = \{t : firstl \leq t \leq lastl\}$

$TO = \{t : firstl - m \leq t < firstl\}$
for each candidate **column** {
  if (**column** indexed by $i \ni T_i \subseteq TRL$) {
    activate **column**
    activate all rows associated with **column**
  }
  if (**column** indexed by $t \in TO$) {
    activate **column**
    change **column**'s objective coefficient to 0
    for each **row** associated with **column** {
      if ((**row**'s sense is " $\leq$ ") and (all **row**'s coefficients ≥ 0)) {
        activate **row**
      }
      if (**row** indexed by $t \in TRL$) {
        activate **row**
      }
      if ((**row** indexed by i) and (**row**'s sense is " $=$ ")) {
        activate **row**
        change **row**'s sense to "$\leq$"
      }
    }
  }
  if (**column** indexed by $t \in TRL$) {
    activate **column**
    for each **row** associated with **column** {
      if (**row** is relaxed) {
        add **row**'s dual multiplier to **column**'s objective coefficient
        if (**row** not indexed by $t > lastl$) {
          activate **row**
          if (**row** is not indexed by t) {
            change **row**'s sense to "$\leq$")
          }
        }
      }
      else {
        activate **row**
      }
    }
  }
}
solve subproblem
add objective value to *lower_bound*
$\ell = \ell + 1$
}

```
for each relaxed row called row {
   add row's dual multiplier · row's RHS to lower_bound
}
report lower_bound as the Lagrangian cascade objective value
}
```

3. Parameter Selection

a. Selection of Lagrangian Cascade Width, *lwid*

Lagrangian cascade solution quality should tend to improve as *lwid* increases. Additionally, knowledge of the problem being solved may be useful when selecting *lwid*. Prior insight as to where the dual multipliers have small absolute values may allow selection of rows that can be relaxed without significantly altering the objective function value. In an extreme (and unrealistic) case, *lwid* might be chosen so that all the Lagrange-relaxed rows have optimal monolith solution multipliers of zero, suggesting that the problem instance is effectively separable. Since prior knowledge of where weak duals occur is not likely, this topic is not pursued further.

A related issue regards selecting *lwid* based on the prior selection of *caswid* and v. Proximal and Lagrangian cascade subproblems whose time periods roughly coincide require that multipliers from the beginning and end of a proximal cascade subproblem be used by the Lagrangian cascade subproblem. Alternatively, choosing *lwid* so as to avoid alignment of $firstl$ and $firstp$ exploits the conjecture that the values of the multipliers may be more likely to resemble the monolith-optimal ones far from the ends of a proximal cascade subproblem.

As with a proximal cascade, a larger Lagrangian cascade width results in fewer subproblems and fewer Lagrange-relaxed rows. But, unlike the proximal cascade, the Lagrangian cascade solution cannot produce a weaker bound when two subproblems are merged by activating the intervening Lagrange-relaxed rows. Thus, two Lagrangian cascade subproblems should always be merged if problem size allows. To show this, define $TR' \subseteq \cup_\ell TR^\ell$ to be a subset of the relaxed periods of the Lagrangian cascade LC given

in Chapter II. Further, define problem LC' (with solution value $Z^{LC'}$) to be a Lagrangian relaxation of the same form as LC but with only the staircase periods for $t \in TR'$ Lagrange-relaxed. Then, the following relationship holds between the solutions of LC', LC, and B :

Theorem 3.1 $\quad Z^B \geq Z^{LC'} \geq Z^{LC}$.

Proof: The left-hand inequality is immediate by the theorem of weak Lagrangian duality.

The right-hand inequality also follows using weak Lagrangian duality, as well as the relationship $TR' \subseteq \cup_\ell TR^\ell$:

$$
\begin{aligned}
Z^{LC'} = \quad & \min \quad \sum_{i \in I} h_i P_i \\
& \qquad + \sum_{i \in I} \alpha_i \left(d_i - \sum_{t \in T_i} X_{it} - P_i \right) \\
& \qquad + \sum_{t \in TR'} \beta_t \left(s_t - \sum_{i \in I} \sum_{t' \in T_i \cap TS_t} a_{itt'} X_{it'} \right) \\
& \quad s.t. \quad \sum_{i \in I} \sum_{t' \in T_i \cap TS_t} a_{itt'} X_{it'} \leq s_t \qquad \forall t \notin TR' \\
& \qquad\quad X_{it}, P_i \geq 0 \;\; \forall i \in I \, , t \in T \\
\geq \quad & \min \quad \sum_{i \in I} h_i P_i \\
& \qquad + \sum_{i \in I} \alpha_i \left(d_i - \sum_{t \in T_i} X_{it} - P_i \right) \\
& \qquad + \sum_{t \in (\cup_\ell TR^\ell)} \beta_t \left(s_t - \sum_{i \in I} \sum_{t' \in T_i \cap TS_t} a_{itt'} X_{it'} \right) \\
& \quad s.t. \quad \sum_{i \in I} \sum_{t' \in T_i \cap TS_t} a_{itt'} X_{it'} \leq s_t \qquad \forall t \notin \cup_\ell TR^\ell \\
& \qquad\quad X_{it}, P_i \geq 0 \;\; \forall i \in I \, , t \in T \\
= \quad & Z^{LC}.
\end{aligned}
$$

□

Thus, combining Lagrangian cascade subproblems cannot reduce solution quality. Moreover, combining subproblems results in fewer relaxed rows, so solution quality should usually improve.

b. Selection of Dual Multipliers

The similarity of the Lagrange-relaxed row penalties to the monolith's optimal dual multipliers is key to the quality of a Lagrangian cascade objective value. This similarity is a function of 1) the overall quality of the corresponding proximal cascade, 2) which proximal cascade subproblem is chosen to provide the multipliers, and 3) the appropriateness of any modifications made to the multipliers. Improving the quality of the proximal cascade is discussed earlier. In this section we discuss how to choose the best, or best combination of subproblems from which to select multipliers. We also discuss what modifications to these multipliers might improve the Lagrangian bound.

We offer two methods for selecting the proximal cascade subproblem from which a Lagrangian multiplier is chosen. The *finalper* method chooses the multiplier associated with the last subproblem in which the corresponding row is active. For example, a demand row associated with columns indexed by periods 5 through 25 may be active in numerous subproblems. Selecting the last subproblem in which period 25 is active may give the best representation of the difficulty required to satisfy that particular demand. Since earlier subproblems include only a limited number of time periods to meet the demand, the Lagrange multiplier may reflect an exaggerated marginal cost of constraint satisfaction.

A more promising technique of multiplier selection averages the dual multipliers from all subproblems in which a row is active, weighted by the number of periods active in that row. This strategy allows the dual variable to reflect a temporal sampling of the resource costs involved in satisfying that constraint. This method, *avgper*, provides better bounds in the case study, and is used throughout.

A potential difficulty of multiplier selection involves the situation where *finalper* or *avgper* computes a penalty for a Lagrange-relaxed demand row that is zero or close to zero. The resulting incentive for a Lagrangian cascade to satisfy demand is negligible—an unlikely situation if the optimal multiplier is used. This discrepancy is redressed by employing a pair of heuristic parameters, *minfrac* and *mdmin*. *minfrac* specifies a threshold fraction for a demand row multiplier, *mdmin* specifies the modified amount. If any demand row multiplier is less than *minfrac* multiplied by the average of the demand

multipliers, its value is reset to *mdmin* times that average. This technique is used throughout the case study.

4. Desirable Model Characteristics for the Lagrangian Cascade

Model characteristics that affect the solution quality of a proximal cascade can be expected to have a similar affect on the solution quality of a Lagrangian cascade. Minimizing the number of Lagrange-relaxed rows is paramount; hence, smaller staircase overlaps should be better. Small row widths also reduce the number of relaxations. Finally, models that include any side constraints whose columns force a relaxation (by being active in more than one subproblem) may often reduce the solution quality of a Lagrangian cascade.

C. SUMMARY

This chapter details the proximal and Lagrangian cascade heuristics using a simple model structure, the multi-commodity elastic-demand staircase linear program. These methods must be implemented on a complex model to be of real use. Consequently, the next chapter is devoted to applying the cascades to the problem motivating the dissertation, namely, the Air Force mobility problem as modelled by the NPS/RAND Mobility Optimizer.

IV. THE NPS/RAND MOBILITY OPTIMIZER

A. INTRODUCTION

The NPS / RAND Mobility Optimizer (NRMO) is under development as an alternative and compliment to simulation for USAF strategic airlift analysis. Designed in the summer of 1996, it is the consolidation of mobility optimization models from NPS [Morton, Rosenthal, and Lim, 1995] and RAND [Killingsworth and Melody, 1994]. The project's sponsor is the USAF Studies and Analyses Agency, Global Mobility Branch.

Strategic airlift is defined as: "...the movement of units, personnel and material in support of all Department of Defense agencies between the continental United States and overseas areas" [US Air Force, 1992, p. 301]. Although this definition embodies many missions, a primary goal of strategic airlift is to maximize the on-time delivery of combat and support forces to any foreign region specified by the national command authorities. NRMO represents strategic airlift as a multi-period, multi-commodity network-based LP with a large number of side constraints. A model instance provides insight into mobility issues such as aircraft fleet and infrastructure adequacy, as well as the identification of system bottlenecks. Multiple scenarios may be used to address questions of fleet selection and airfield improvements.

There are four primary input requirements of the NRMO LP: 1) the required cargo and passenger movements as delineated by the Time Phased Force Deployment Document (TPFDD), a widely used planning database, 2) the types and numbers of available aircraft and crews, 3) the usable airfields, and 4) the allowable routes for each aircraft type. The LP minimizes the weighted sum of late and undelivered cargo penalties, subject to restrictions such as aircraft flow balance, aircraft payload, and airfield capacity. The solution specifies the airlift mission assignments by requirement moved, aircraft and route flown, and time delivered. From this output, information such as unit closure (the time when all of a unit's cargo and passengers have been delivered) may be computed. Return routings and

airfield saturation levels are also given in the LP solution, as well as the marginal values of resources.

In addition to the four primary inputs, other data allow NRMO to model aerial refueling, geographic crew movement, and intra-theater airlift. If directed by the scenario input, NRMO can assign dual-role aircraft as either airlifters or aerial refueling tankers, and reassign them as the contingency warrants. The movement of crews can be modelled geographically by balancing their flow through selected rest bases, and observing overall limits on their number. Finally, NRMO allows intra-theater activity by alternating selected aircraft between tactical and strategic roles, again as the contingency warrants.

NRMO is a very complex example of a multi-commodity, elastic demand-staircase model. With some modifications and additional assumptions, it provides a good case study for cascades. This chapter develops the case study by presenting the model, and then states the proximal and Lagrangian cascade formulations. The monolith formulation of NRMO follows [Rosenthal *et al.* 1997].

B. NRMO FORMULATION

1. Explanation of Terms and Acronyms

The following is a list of terms and acronyms used by the NRMO formulation. As necessary, these terms are explained in greater detail throughout the formulation.

acft	aircraft
APOD	Aerial Port of Debarkation
APOE	Aerial Port of Embarkation
AR	Aerial Refueling
backchannel	returning an empty aircraft from an APOD to an APOE
bed down	resting location of tanker or intra-theater aircraft
cargo types:	bulk - palletized
	oversize - typically vehicles
	outsize - typically tanks or helicopters
	pax - passengers
chop	assignment of aircraft to an intra-theater role
CONUS	Continental United States
CRAF	Civil Reserve Air Fleet, airliners contracted for military service

crew stage	location where aircraft get a fresh aircrew
divert	routing of an intended AR mission that failed
FOB	Forward Operating Base
line id	delivery requirement
MOG	Maximum On Ground, an airfield's capacity
quick turn	unloading an aircraft in theater without servicing
recovery	eventual servicing location of a quick turn mission
RON	Remain Over Night
ston	short ton (2000 lbs.), as opposed to metric ton (1000 kg.)
shuttle	intra-theater mission
super node	an aggregation of APODS to reduce the number of variables
tanker cloud	modeling construct to reduce the number of variables
theater	region of the world where the deliveries occur
ute	utilization

2. Sets

T	time periods
TW_i	delivery time window for line id i
T_u	a set of time periods over which an aircraft's flying hours are limited
FT	flow time periods f = {1,..., maximum mission time }, used for flight times
U	the set of time blocks that limit an aircraft's flying hours
I	line ids
I_{fob}	subset of line ids whose destination is a FOB
I_{apd}	subset of line ids whose destination is an APOD.
$I_{b,dst}$	subset of line ids that have base b (FOB or APOD) as a destination
$I_{b,trn}$	subset of line ids that have APOD b as a transshipment node
$I_{b,sup}$	subset of line ids that use super node b
C	cargo types $\{bulk, over, out, pax\}$
CC	cargo types $\{bulk, over, out\}$
C_a	subset of cargo types that can be carried by acft a
A	set of acft types
A_c	subset of acft types that can carry cargo type c
A_{mix}	subset of acft types that can carry pax and at least one other cargo type ($bulk$, $over$, or out)
A_{pax}	subset of acft that can carry passengers
A_{tkr}	subset of tanker acft types
A_{rfl}	subset of acft that can be refueled by a tanker
A_{chp}	subset of acft that can be "chopped", *i.e.,* assigned to the theater

B	set of all "bases" (APOE, APOD, FOB, super, enroute, waypoint, bed down, and aerial refueling points)
B_{sup}	subset of bases that are super nodes
B_{fob}	subset of bases that are FOBs
B_e	subset of bases that are embarkation nodes
B_{arp}	subset of bases that are AR points
B_{tkr}	subset of bases that are bed-down bases for tankers
BS_{rec}	set of super nodes that have at least one recovery base
B_{way}	set of bases that are enroute navigational waypoints
$BS_{b,dwn}$	set of super nodes that have b as the shuttle bed-down node
$BF_{b,sup}$	set of FOB's that call b their super node plus the super node itself
$BA_{b,tkr}$	subset of B_{arp} that are served by $b \in B_{tkr}$
$BT_{b,arp}$	subset of B_{tkr} that serve $b \in B_{arp}$
B_{crw}	crew stage bases
R	routes
RD	delivery routes
RB	backchannel routes
RB_{rec}	subset of backchannel routes that include a recovery base
RD_b	delivery routes that use base b (terminal node is a super, not FOB or APOD)
$RD_{ia,dir}$	subset of routes that can be flown by a and carry i for direct delivery
$RD_{ia,trn}$	subset of routes that can be flown by a and carry i for transshipment
RB_{ab}	subset of backchannel routes that use b and can be flown by a
$RD_{b,div}$	set of delivery routes that have b as a divert base
$RB_{b,div}$	set of backchannel routes that have b as a divert base
$R_{b,ori}$	routes whose origin is base b
$R_{b,dst}$	routes whose destination is base b

3. Data

Mission time data

$rtrv_{ar}$	total travel time for acft a to travel on route r (periods)
trv_{ar}	rounded $rtrv_{ar}$ (integer periods)
$retrv_{abr}$	travel time for acft a to reach base b when flying route r (periods)
$etrv_{abr}$	rounded $retrv_{abr}$ (integer periods)
$maxtrv_a$	maximum total travel time along any route for acft a (integer periods)
$msntime_{arf}$	time flown f periods into a mission (hrs) • $hrsper$ if $rtrv_{ar} > f$ (mission continues throughout fth period) • 0 if $rtrv_{ar} < f - 1$ (mission teminates before fth period) • $hrsper \cdot (rtrv_{ar} - (f-1))$ if $f - 1 \leq rtrv_{ar} \leq f$ (mission terminates during f'th period)

$ttrv_{ab}$	rounded $rttrv_{abr}$ (integer periods)
$tkrtime_{abb'}$	in-flight time for tanker a flying from b to b' and back (hrs)
$tkrrate_{abb'}$	maximum number of tanker shuttles to AR point b' per period for tanker a when it is bedded at b
$shutrate_{ai}$	maximum number of in-theater shuttles per aircraft per period
$sgtime_{ab}$	ground time for shuttle aircraft a at base b (hrs)
$gtrv_i$	in-theater ground travel time for i (periods)
$shuttime_{ia}$	in-flight shuttle time (hrs)
$flttime_{arf}$	same as $msntime_{arf}$, but only includes flying time thus, $flttime_{arf} < msntime_{arf}$, since all missions have some ground time
$gtime_{abr}$	ground time for aircraft a at base b when flying route r (hrs)
$qtime_{abr}$	offload time only for acft a at base b when flying route r with recovery used (hrs)
$ctrv_{abr}$	travel time to b, plus crew rest, for a along r (integer periods)
$cttrv_{ab}$	$ttrv_{ab}$ plus crew rest (integer periods)
$dhtrv_{b'b}$	travel time for deadheading crew from b' to b (integer periods)
$rttrv_{ab}$	tanker a reposition time (approx 2 days) from embarkation or bed-down base b to cloud (integer periods)
Aircraft data	
$newac_{at}$	number of new acft of type a available in period t
$cumac_{at}$	$= \sum_{t' \leq t} newac_{at'}$
$crewrat_a$	ratio of available crews to acft a
$purecap_{iac}$	number of stons of unit i's cargo of type c that can be loaded on acft a for a 3200nm flight
$maxpax_a$	maximum number of pax that can be loaded on an acft of type a
$paxfrac_a$	fraction of an acft's capacity that can be loaded with pax
$rangefac_{iar}$	fraction of acft available for loading when flying route r for line id i
$restrew_a$	unit reward for resting acft a at base $b \in B_e$ $(\max_{ic}\{purecap_{iac}\} \cdot latepen_i \cdot 0.01)$
$usepen_a$	usage penalty for theater aircraft and tanker reassignments
$dhpen_a$	penalty for deadheading crews
$tkreqvs_{abr}$	amount of a full tanker consumed by acft a refueling at AR b on r (KC10 equiv)
$tkrprop_{abb'}$	proportion of a full tanker (KC10 equiv) available when a is a refueler at AR base b' and is bedded at b
$dpct_a$	fraction of AR attempts by receiving acft a that fail
$urate_a$	number of hours per day that aircraft a can fly
$initchop_{ab}$	initial acft chopped to theater

Movement requirements data

rdd_i	required delivery date
dem_{ic}	stons of demand for line id i of type c
$latepen_i$	late penalty (delivered after rdd_i) for i per day per ston
$maxlate_i$	maximum number of time periods late a delivery for line id i can arrive
$nogopen_i$	non-delivery penalty per ston ($\geq latepen_i \cdot maxlate_i$)

Other data and notational conventions

$hrsper$	number of hours per period
$acpkg_{ab}$	unit mog consumption of aircraft a at base b
$mogeff_b$	mog efficiency at b
mog_b	airfield capacity: service spot hours per period at b
$I(\cdot)$	1 if argument is true; 0 otherwise.
$(x)^+$	$= \max\{0, x\}$
$\bar{S}$	complement of a set S
$\setminus$	set difference, *i.e.*, $S \setminus T = S \cap \overline{T}$

In general, constraints and variables are assumed to exist
only for the appropriate combinations of their indices

4. Decision Variables

Aircraft mission variables

XD_{iart}	# of aircraft a *direct* delivering i on route r departing at time t
XT_{iart}	# of aircraft a delivering a *transshipment* load (from APOE to a transshipment APOD) of i on route r departing at time t
XDR_{iart}	# of aircraft a *direct* delivering i on *quick turn* route r departing at time t
XTR_{iart}	# of aircraft a delivering a *transshipment* load of i on *quick turn* route r departing at time t. The load is shuttled after transshippment,
XS_{iat}	# of (round trip) shuttle missions of type acft a delivering i in t
Y_{art}	# of aircraft a recovering on route r departing at t
$TKRA_{abb't}$	# of tanker sorties of type a flown from $b \in B_{tkr}$ to $b' \in B_{arp}$ in t

Aircraft inventory variables

RON_{abt}	number (#) of acft of type a Remaining Over Night at $b \in B_e$ in t
$RONT_{abt}$	# of acft of type a "RONing" without recovery in t
$RONR_{abt}$	# of acft of type a "RONing" with recovery in t
$IRONT_{ab}$	# of acft of type a initially assigned to b (non-recovery)
$IRONR_{ab}$	# of acft of type a initially assigned to b (recovery)
$THCHOP_{abt}$	# of acft assigned to super b's shuttle fleet from non-recovery routes in t
$THCHOPR_{abt}$	# of acft assigned to super b's shuttle fleet from recovery routes in t
$TKRB_{abt}$	# of tankers a whose bed-down base is $b \in B_{tkr}$ in t

Aircraft changing roles

$ALLOC_{abt}$	# of new acft a allocated to $b \in B_e$ in t
$TKREC_{abt}$	# of tankers a leaving $b \in B_e$ in t for service as a refueler (for cloud)
$TKRCE_{abt}$	# of tankers a leaving tanker fleet (cloud) in t for $b \in B_e$ for cargo hauling
$TKRBC_{abt}$	# of tankers a leaving $b \in B_{tkr}$ in t for reassignment or service as a cargo hauler
$TKRCB_{abt}$	# of tankers a being reassigned (from cloud) in t to $b \in B_{tkr}$ for refueling

Cargo

$DTONS_{iact}$	stons of i's cargo of type c direct delivered by a that will arrive in t.
$TTONS_{iact}$	stons of i's cargo of type c for transshipment by a arriving at (the transshipment node) in t
$STONS_{iact}$	stons of i's cargo of type c shuttled by a in t
$GTONS_{ict}$	stons of i's cargo of type c ground that will arrive at the FOB in t Note: when indexed by "pax," $DTONS, TTONS, STONS,$ and $GTONS$ represents number, not stons, of pax.
$NOGO_{ic}$	stons of i's cargo of type c not delivered

Crews

$SCREWS_{abt}$	# of strategic airlift crews available (rested) for a at $b \in B_{crw}$ at the beginning of time t
$DHCREWS_{ab'bt}$	# of deadheading crews for a leaving b' at time t for reassignment to b

5. Formulation

OBJ: Objective function

minimize

$$\sum_{i\in I}\sum_{a\in A}\sum_{c\in C_a,}\sum_{t\in TW_i} latepen_i \cdot (t - rdd_i)^+ \cdot DTONS_{iact}$$

$$+ \sum_{i\in I_{fob}}\sum_{a\in A}\sum_{c\in C_a}\sum_{t\in TW_i} latepen_i \cdot (t - rdd_i)^+ \cdot STONS_{iact}$$

$$+ \sum_{i\in I_{fob}}\sum_{c\in C}\sum_{t\in TW_i} latepen_i \cdot (t - rdd_i)^+ \cdot GTONS_{ict}$$

$$+ \sum_{i\in I}\sum_{c\in C} nogopen_i \cdot NOGO_{ic}$$

$$+ \sum_{a\in A_{chp}}\sum_{b\in B_{sup}}\sum_{t\in T} usepen_a \cdot [THCHOP_{abt} + THCHOPR_{abt}]$$

$$+ \sum_{a\in A_{tkr}}\sum_{b\in B_e}\sum_{t\in T} usepen_a \cdot TKREC_{abt} + \sum_{a\in A_{tkr}}\sum_{b\in B_{tkr}}\sum_{t\in T} usepen_a \cdot TKRBC_{abt}$$

$$- \sum_{a\in A}\sum_{b\in B_e}\sum_{t\in T} restrew_a \cdot RON_{abt} + \sum_{a\in A}\sum_{b,b'\in B_{crw}}\sum_{t\in T} dhpen_a \cdot DHCREW_{abb't}$$

Minimize the sum of: 1) late penalty · number of days late · late cargo delivered directly to the line id's destination; 2) late penalty · number of days late · late cargo shuttled (from the transshipment base) to the line id's destination; 3) late penalty · number of days late · late cargo delivered by ground from the transshipment base; 4) nondelivery penalty · undelivered cargo; 5) usage penalty · number of chopped aircraft or reassigned tankers; 6) a small reward (negative penalty) · number of aircraft remaining overnight at an APOE (often CONUS, and thereby near home station); and 7) crew deadhead penalty · deadheading crews.

ACBALE: aircraft balance at embarkation nodes

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XT_{iart} + \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XD_{iart}$$

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XTR_{iart} + \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XDR_{iart}$$

$$+I(a \in A_{tkr}) \cdot [TKREC_{abt}] + RON_{abt} = RON_{abt-1} + \sum_{r \in RB_{ab}} Y_{art-trv_{ar}}$$

$$+ALLOC_{abt} + I(a \in A_{tkr}) \cdot [TKRCE_{abt}] \qquad \forall a \in A, b \in B_e, t \in T$$

AirCraft BALance at apoE's: For each aircraft type, APOE, and time period (day); departing transshipment missions + departing direct delivery missions + assignments to tanker duty (if aircraft is a tanker) + overnight resting aircraft = resting aircraft from yesterday + arriving backchannel missions + newly assigned aircraft + reassignments from tanker duty (if aircraft is a tanker). Note that direct delivery missions and transshipment missions can be selected to recover away from the APOD (XDR, XTR) or recover at the APOD (XD, XT) missions. This is true throughout the formulation, except as noted.

ACBALSUP: aircraft balance at SUPER debarkation nodes

$$\sum_{r \in RB_{ab} \cap \bar{R}\bar{B}_{rec}} Y_{art} + RONT_{abt} + THCHOP_{abt} =$$

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XT_{iart-trv_{ar}} + \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XD_{iart-trv_{ar}}$$

$$+RONT_{abt-1} + THCHOP_{abt-1} + I(t = 1) \cdot IRONT_{ab} \qquad \forall a \in A, b \in B_{sup}, t \in T$$

AirCraft BALance at SUPer's: A "super" node is a surrogate for all bases in the theater. Flow balance is done with supers, but MOG is constrained at the actual theater APODs and FOBs. Additionally, this constraint only addresses missions that recover at the APOD. Other missions are constrained in ACBALREC. For each aircraft type, super, and time period, the departing backchannel missions + overnight resting aircraft + total aircraft chopped to the theater = arriving transshipment missions + arriving direct delivery missions

(for those line ids whose destination is an APOD) + last night's resting aircraft + yesterday's total of chopped aircraft + the initial "chops" to theater (if it is the first time period).

ACBALREC: aircraft balance at SUPER debarkation nodes with recovery

$$\sum_{r \in RB_{ab} \cap RB_{rec}} Y_{art} + RONR_{abt} + THCHOPR_{abt} =$$

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XTR_{iart-trv_{ar}} + \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XDR_{iart-trv_{ar}}$$

$$+RONR_{abt-1} + THCHOPR_{abt-1} + I(t=1) \cdot IRONR_{ab} \qquad \forall a \in A, b \in BS_{rec}, t \in T$$

AirCraft BALance at supers using RECovery routes: Same as ACBALSUP, but balances flow for missions not recovering at the APOD.

INITIRON: allocate initial theater assignments

$$IRONT_{ab} + IRONR_{ab} = initchop_{ab} \quad \forall a \in A_{chp}, b \in B_{sup}$$

INITIal RONs in theater: For period 1 and all aircraft and supers; the sum of RONS at APOD recoveries plus the RONS at non-APOD recoveries equals the initial aircraft chopped to theater.

ACALLOC: allocate newly available aircraft

$$\sum_{b \in B_e} ALLOC_{abt} = newac_{at} \quad \forall a \in A, t \in T$$

AirCraft ALLOCation: For each aircraft type and time period; the sum of all new allocations to APOE's = the amount newly available.

SHUTLBND: don't send more shuttles than available

$$\sum_{i \in I_{b,sup} \cap I_{fob}} \frac{XS_{iat}}{shutrate_{ai}} \leq [THCHOP_{abt} + THCHOPR_{abt}] \quad \forall a \in A, b \in B_{sup}, t \in T$$

SHUTtLe BouND: For each aircraft type, "super" APOD, and time period; the number of round trip shuttle missions divided by the daily number of round trip missions per aircraft $\leq$ the total chopped aircraft in the theater.

TKRBND: don't use more tankers than available

$$\sum_{b' \in BA_{b,tkr}} \frac{TKRA_{abb't}}{tkrrate_{abb'}} \leq TKRB_{abt} \quad \forall a \in A_{tkr}, b \in B_{tkr}, t \in T$$

TanKeR BouND: For all tankers, tanker bed down bases, and time periods: the number of AR sorties flown to all tracks divided by the daily sortie rate $\leq$ tankers assigned to the bed down base.

CLOUDBAL: flow balance: leaving and entering tanker fleet

$$\sum_{b \in B_e} TKREC_{abt-ttrv_{ab}} + \sum_{b \in B_{tkr}} TKRBC_{abt-ttrv_{ab}} =$$
$$\sum_{b \in B_e} TKRCE_{abt} + \sum_{b \in B_{tkr}} TKRCB_{abt} \qquad \forall a \in A_{tkr}, t \in T$$

tanker CLOUD BALance: The "tanker cloud" is a node at which, as a modeling convenience, we assume role changes take place for multi-role aircraft that can be tankers or airlifters. The "cloud" serves as a control point that reduces the number of required assignment and de-assignment variables. For all tanker aircraft types and time periods: newly assigned tankers from all APOEs (adjusted for travel time) + newly de-assigned tankers from all tanker bed down bases (also adjusted for travel time) = tankers returning to all APOEs + tankers deploying to all bed down bases. Note that de-assigning a tanker from

a bed down base does not force it back to an APOE; it could be re-assigned to another bed down base.

TKRINVT: tanker inventory at tanker bed-downs

$$TKRBC_{abt} + TKRB_{abt} = TKRCB_{abt} + TKRB_{abt-1} \quad \forall a \in A_{tkr}, b \in B_{tkr}, t \in T$$

TanKeR INVenTory: For all tanker aircraft types, tanker bed down bases, and time periods; newly de-assigned tankers + total tankers assigned = newly assigned tankers + total tankers assigned from last period.

ARMOG: aerial refueling capacity constraint

$$\begin{aligned}
&\sum_{i \in I} \sum_{a \in A_{rfl}} \sum_{r \in RD_b \cap RD_{ia,dir}} tkreqvs_{abr} \cdot XD_{iart-etrv_{abr}} \\
&+ \sum_{i \in I} \sum_{a \in A_{rfl}} \sum_{r \in RD_b \cap RD_{ia,trn}} tkreqvs_{abr} \cdot XT_{iart-etrv_{abr}} \\
&\sum_{i \in I} \sum_{a \in A_{rfl}} \sum_{r \in RD_b \cap RD_{ia,dir}} tkreqvs_{abr} \cdot XDR_{iart-etrv_{abr}} \\
&+ \sum_{i \in I} \sum_{a \in A_{rfl}} \sum_{r \in RD_b \cap RD_{ia,trn}} tkreqvs_{abr} \cdot XTR_{iart-etrv_{abr}} \\
&+ \sum_{a \in A_{rfl}} \sum_{r \in RB_{ab}} tkreqvs_{abr} \cdot Y_{art-etrv_{abr}} \\
&\leq \sum_{b' \in BT_{b,arp}} \sum_{a \in A_{tkr}} tkrprop_{ab'b} \cdot TKRA_{ab'bt} \qquad \forall b \in B_{arp}, t \in T
\end{aligned}$$

Air Refueling MOG: Despite the apparent contradiction of terms, this constraint is the air refueling analog to airfield MOG — it constrains the capacity of an AR track. For all air refueling points and time periods; the fuel required by direct delivery, transshipment, and backchannel missions hitting the track in this time period $\leq$ the amount of fuel available from tanker sorties flown to the track.

UTE: utilization rate

$$\sum_{t\in T_u}\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} flttime_{arf}\cdot XD_{iart-(f-1)}$$
$$+\sum_{t\in T_u}\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} flttime_{arf}\cdot XT_{iart-(f-1)}$$
$$+\sum_{t\in T_u}\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} flttime_{arf}\cdot XDR_{iart-(f-1)}$$
$$+\sum_{t\in T_u}\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} flttime_{arf}\cdot XTR_{iart-(f-1)}$$
$$+\sum_{i\in I_{fob}}\sum_{t\in T_u} shuttime_{ia}\cdot XS_{iat}+\sum_{t\in T_u}\sum_{r\in RB_b}\sum_{f\in FT} flttime_{arf}\cdot Y_{art-(f-1)}$$
$$+I(a\in A_{tkr})\cdot\Bigg[\sum_{b\in B_{tkr}}\sum_{b'\in B_{arp}}\sum_{t\in T_u} tkrtime_{abb'}\cdot TKRA_{abb't}$$
$$+\sum_{b\in B_e}\sum_{t\in T_u} hrsper\cdot rttrv_{ab}\cdot TKREC_{abt}$$
$$+\sum_{b\in B_{tkr}}\sum_{t\in T_u} hrsper\cdot rttrv_{ab}\cdot TKRBC_{abt}\Bigg]$$
$$\leq\sum_{t\in T_u} cumac_{at}\cdot urate_a \qquad \forall a\in A, u\in U$$

UTilization ratE: Sums all varieties of flight time, so the left-hand-side parameters of this constraint accumulates flight time only of missions operating during blocks of UTE rate enforcement. The utilization rate blocks B in NRMO are defined arbitrarily. They are motivated by the fact that over a period of several weeks, an aircraft can historically fly an ill defined average amount of time. Thus, UTE rate blocks are generally between 20 and 30 days.

For each aircraft type and UTE rate block; the flight time of all direct, transshipment, shuttle, and backchannel missions (as well as deployed and deploying tankers, if appropriate) $\leq$ total aircraft hours available $\cdot$ maximum hours per day of average aircraft utilization. The f index corresponds to the number of days into a mission, so when $f = 1$, a typical term is the flight time of a mission's first day times the number of missions (of that type)

launched that day. Similarly, when $f = 2$, a typical term corresponds to the flight time of a mission's second day times the number of missions (of that type) launched on the previous day.

ACCONSUME: max acft usage to lessen rounding effects

$$\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} msntime_{arf}\cdot XD_{iart-(f-1)}$$
$$+\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} msntime_{arf}\cdot XT_{iart-(f-1)}$$
$$\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} msntime_{arf}\cdot XDR_{iart-(f-1)}$$
$$+\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} msntime_{arf}\cdot XTR_{iart-(f-1)}$$
$$+\sum_{i\in I_{fob}}\frac{hrsper}{shutrate_{ai}}\cdot XS_{iat}+\sum_{r\in RB}\sum_{f\in FT} msntime_{arf}\cdot Y_{art-(f-1)}$$
$$+I(a\in A_{tkr})\cdot\left[\sum_{b\in B_{tkr}}\sum_{b'\in B_{arp}}\frac{hrsper}{tkrrate_{abb'}}\cdot TKRA_{abb't}\right.$$
$$+\sum_{b\in B_e} rttrv_{ab}\cdot hrsper\cdot TKREC_{abt}$$
$$\left.+\sum_{b\in B_{tkr}} rttrv_{ab}\cdot hrsper\cdot TKRBC_{abt}\right]$$
$$+\sum_{b\in B_e} hrsper\cdot RON_{abt}+\sum_{b\in B_{sup}} hrsper\cdot[RONT_{abt}+RONR_{abt}]$$
$$\leq hrsper\cdot cumac_{at} \qquad \forall a\in A, t\in T$$

AirCraft CONSUMEd: Structurally similar to UTE, this constraint reduces the effect of time discretization. It supplements the flow balance constraints, which may deal with short missions whose rounded duration is 0 periods. For all aircraft types and time periods; mission time of all direct, transshipment, shuttle, and backchannel missions (as well as deployed and deploying tankers, if appropriate) plus resting aircraft $\leq$ total aircraft hours available.

DCAPACITY: aircraft capacity for direct delivery

$$\sum_{c \in C_a \cap CC} \frac{DTONS_{iact}}{purecap_{iac}} + \frac{paxfrac_a \cdot DTONS_{i,a,pax,t}}{maxpax_a} \cdot I(a \in A_{pax})$$
$$\leq \sum_{r \in RD_{ia,dir}} rangefac_{iar} \cdot [XD_{iart-trv_{ar}} + XDR_{iart-trv_{ar}}] \quad \forall i \in I, a \in A, t \in TW_i$$

Direct delivery mission CAPACITY: For each line id, aircraft type, and time period; the number of tons delivered (summed over cargo classes) divided by the aircraft capacity by cargo type and unit + the passengers delivered divided by the passenger capacity $\leq$ the number of missions launched in support of i by aircraft of type a along any route, launched long enough ago so as to be arriving at time t. $paxfrac$ specifies the portion of the aircraft filled if fully loaded with passengers. Parameter $rangefac$ is frequently 1, but is reduced if a leg of route r is long enough to exceed the aircraft's range-payload performance.

TCAPACITY: aircraft capacity for transshipments

$$\sum_{c \in C_a \cap CC} \frac{TTONS_{iact}}{purecap_{iac}} + \frac{paxfrac_a \cdot TTONS_{i,a,pax,t}}{maxpax_a} \cdot I(a \in A_{pax})$$
$$\leq \sum_{r \in RD_{ia,trn}} rangefac_{iar} \cdot [XT_{iart-trv_{ar}} + XTR_{iart-trv_{ar}}] \quad \forall i \in I_{fob}, a \in A, t \in TW_i$$

Transshipment mission CAPACITY: Same as DCAPACITY, but applies to missions flown in support of cargo and pax deliveries to transshipment APODs (for subsequent transshipment).

SCAPACITY: aircraft capacity for shuttle deliveries

$$\sum_{c \in C_a \cap CC} \frac{STONS_{iact}}{purecap_{iac}} + \frac{paxfrac_a \cdot STONS_{i,a,pax,t}}{maxpax_a} \cdot I(a \in A_{pax})$$
$$\leq srange_{ia} \cdot XS_{iat} \qquad \forall i \in I_{fob}, a \in A, t \in TW_i$$

Shuttle mission CAPACITY: Same as DCAPACITY and SCAPACITY, but applies to intra-theater missions moving cargo from transshipment APODs to FOBs.

DPAXCAP: aircraft capacity for direct delivery of pax

$$DTONS_{i,a,pax,t} \leq \sum_{r \in RD_{ia,dir}} maxpax_a \cdot [XD_{iart-trv_{ar}} + XDR_{iart-trv_{ar}}] \quad \forall i \in I, a \in A_{mix}, t \in TW_i$$

Direct delivery mission PAX CAPacity: For each line id, aircraft type, and time period; the number of pax moved must not exceed the maximum pax per mission · number of missions flown. It supplements DCAPACITY, which would (by itself) allow the aircraft to be fully loaded with pax, despite available seating configurations.

TPAXCAP: aircraft capacity for transshipment of pax

$$TTONS_{i,a,pax,t} \leq \sum_{r \in RD_{ia,trn}} maxpax_a \cdot [XT_{iart-trv_{ar}} + XTR_{iart-trv_{ar}}] \quad \forall i \in I_{fob}, a \in A_{mix}, t \in TW_i$$

Transshipment mission PAX CAPacity: Same as DPAXCAP, but applies to transshipment missions.

SPAXCAP: aircraft capacity for delivery of pax by shuttles

$$STONS_{i,a,pax,t} \leq maxpax_a \cdot XS_{iat} \quad \forall i \in I_{fob}, a \in A_{mix}, t \in TW_i$$

Shuttle mission PAX CAPacity: Same as DPAXCAP and TPAXCAP, but applies to intra-theater shuttle missions.

MEETDEM: meet demand for each line id

$$\sum_{a \in A_c} \sum_{t \in TW_i} DTONS_{iact} + NOGO_{ic}$$
$$+I(i \in I_{fob}) \cdot \left[\sum_{a \in A_c} \sum_{t \in TW_i} STONS_{iact} + \sum_{t \in TW_i} GTONS_{ict} \right] = dem_{ic} \quad \forall i \in I, c \in C$$

MEET DEMand: For each line id and cargo class; direct delivery tons (and pax) moved by all aircraft over the available time window + tons moved by shuttle missions (if destination is a FOB) + tons moved by ground (if destination is a FOB) + cargo NOT moved = demand by unit and cargo class.

TRANSTONS: flow balance for transshipped stons

$$\sum_{a \in A_c} TTONS_{iact} = \sum_{a \in A_c} STONS_{iact}$$
$$+GTONS_{ict+gtrv_i} \cdot I(t + gtrv_i \in TW_i) \quad \forall i \in I_{fob}, c \in C, t \in TW_i$$

TRANSshipment TONS: For each line id, cargo class and time period; Transshipment tons moved from APOE to transshipment APOD by strategic airlift = tons moved from transshipment APOD to FOB by shuttle or ground transport.

INITCREWS: initialize crew placement

$$\sum_{b \in B_{crw}} SCREWS_{abt} + crewrat_a \cdot \sum_{b \in B_{tkr}} TKRB_{abt}$$
$$= crewrat_a \cdot newac_{at} \qquad \forall a, t = 1$$

INITialize CREWS: For all aircraft and time period 1; strategic airlift crews available at all crew stage bases + crew contingent for all pre-deployed tankers = number of crews available.

SCREWBAL: strategic crew balance of flow

$$SCREWS_{abt+1} = SCREWS_{abt}$$

$$+\sum_{i\in I}\sum_{r\in RD_{ia,dir}\cap\bar{R}_{b,ori}}[XD_{iart-ctrv_{abr}} + XDR_{iart-ctrv_{abr}}]$$

$$+\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}\cap\bar{R}_{b,ori}}[XT_{iart-ctrv_{abr}} + XTR_{iart-ctrv_{abr}}]$$

$$+\sum_{r\in RB\cap\bar{R}_{b,ori}} Y_{art-ctrv_{abr}}$$

$$-\sum_{i\in I}\sum_{r\in RD_{ia,dir}\cap\bar{R}_{b,dst}}[XD_{iart-etrv_{abr}} + XDR_{iart-etrv_{abr}}]$$

$$-\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}\cap\bar{R}_{b,dst}}[XT_{iart-etrv_{abr}} + XTR_{iart-etrv_{abr}}]$$

$$-\sum_{r\in RB\cap\bar{R}_{b,dst}} Y_{art-etrv_{abr}}$$

$$+I(b\in B_e)\cdot crewrat_a\cdot[TKRCE_{abt-cttrv_{ab}} - TKREC_{abt}]$$

$$+I(b\in B_{sup})\cdot$$

$$crewrat_a\cdot[THCHOP_{abt-1} - THCHOP_{abt} + I(t=1)\cdot IRONT_{a,b}]$$

$$+I(b\in BS_{rec})\cdot$$

$$crewrat_a\cdot[THCHOPR_{abt-1} - THCHOPR_{abt} + I(t=1)\cdot IRONR_{a,b}]$$

$$+I(b\in B_e, t\neq 1, newac_{at}>0)\cdot crewrat_a\cdot ALLOC_{abt}$$

$$+\sum_{b'\in B_{crw}} DHCREW_{ab'bt-dhtrv_{b'b}} - \sum_{b'\in B_{crw}} DHCREW_{abb't}$$

$$\forall a,b\in B_{crw},\ \ \forall t:(t\in T, t<|T|)$$

Strategic CREW BALance: For all aircraft, crew stage bases, and time periods; the number of crews available tomorrow = the number of crews available today + crews coming out of crew rest from previous direct, transshipment, and backchannel missions - crews required for departing direct, transshipment, and backchannel missions + the net crews made available from tanker deployments and returns (if APOE and tanker aircraft) + the net crews made available from "chopped" and "unchopped" aircraft (if "super" APOD) + new crew allocations + arriving deadhead crews from other bases - deadhead crews departing for other bases.

MOG: airfield capacity

$$\sum_{i\in I}\sum_{a\in A}\sum_{r\in RD_b\cap RD_{ia,dir}} gtime_{abr}\cdot acpkg_{ab}\cdot[XD_{iart-etrv_{abr}}+XDR_{iart-etrv_{abr}}]$$

$$+\sum_{i\in I_{b,dst}}\sum_{a\in A}\sum_{r\in RD_{ia,dir}} gtime_{abr}\cdot acpkg_{ab}\cdot XD_{iart-trv_{ar}}$$

$$+\sum_{i\in I_{b,dst}}\sum_{a\in A}\sum_{r\in RD_{ia,dir}} qtime_{abr}\cdot acpkg_{ab}\cdot XDR_{iart-trv_{ar}}$$

$$+\sum_{i\in I}\sum_{a\in A}\sum_{r\in RD_b\cap RD_{ia,trn}} gtime_{abr}\cdot acpkg_{ab}\cdot[XT_{iart-etrv_{abr}}+XTR_{iart-etrv_{abr}}]$$

$$+\sum_{i\in I_{b,trn}}\sum_{a\in A}\sum_{r\in RD_{ia,trn}} gtime_{abr}\cdot acpkg_{ab}\cdot XT_{iart-trv_{ar}}$$

$$+\sum_{i\in I_{b,trn}}\sum_{a\in A}\sum_{r\in RD_{ia,trn}} qtime_{abr}\cdot acpkg_{ab}\cdot XTR_{iart-trv_{ar}}$$

$$+\sum_{i\in(I_{b,dst}\cap I_{fob})\cup I_{b,trn}}\sum_{a\in A} sgtime_{ab}\cdot acpkg_{ab}\cdot XS_{iat}$$

$$+\sum_{b'\in B_{sup}\cap BS_{b,dwn}}\sum_{a\in A} hrsper\cdot acpkg_{ab}\cdot[THCHOP_{ab't}+THCHOPR_{ab't}]$$

$$+\sum_{a\in A}\sum_{r\in RB_{ab}} gtime_{abr}\cdot acpkg_{ab}\cdot Y_{art-etrv_{abr}}$$

$$+I(b\in B_{tkr})\cdot\left[\sum_{a\in A_{tkr}} hrsper\cdot acpkg_{ab}\cdot TKRB_{abt}\right]$$

$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,dir}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XD_{iart-etrv_{abr}}$$

$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,dir}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XDR_{iart-etrv_{abr}}$$

$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,trn}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XT_{iart-etrv_{abr}}$$

$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,trn}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XTR_{iart-etrv_{abr}}$$

$$+\sum_{a\in A_{rfl}}\sum_{r\in RB_{b,div}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot Y_{art-etrv_{abr}}$$

$$\leq mog_b\cdot mogeff_b \qquad \forall b\in B\setminus B_{sup}\setminus B_{arp}\setminus B_{way}, t\in T$$

Maximum On Ground: For all bases (except super APODs, AR points, and waypoints) and time periods; the aircraft parking required for transiting and terminating direct delivery missions + parking for transiting and terminating transshipment missions + shuttle mission parking (if FOB or transshipment APOD) + chopped aircraft bed down parking (if shuttle bed down base) + backchannel mission parking − parking saved at offload base by using recovery backchannel routes (no fuel or maintenance at offload) + tanker bed down parking (if tanker bed down base) + divert base parking for failed refuelings of direct delivery, transshipment, and backchannel missions $\leq$ available MOG · MOG efficiency.

Non-negativity of all variables.

C. NRMO BY PROXIMAL CASCADE

Much of the NRMO formulation is well suited to a proximal cascade. Depending on the scenario and which features of the airlift system are modelled, the maximum staircase overlap m varies between one and three periods. Travel times for the various missions (accounted for by subtracting the appropriate number of periods from the corresponding variables' subscripts) determine the overlap. If all features are modelled, the maximum crew travel time lag in the $SCREWBAL$ constraint, $\max_{abr}[ctrv_{abr}]+1$, usually determines the maximum staircase overlap. If crews are not modeled, either the maximum mission travel time, $\max_{ar}[trv_{ar}]$, or the maximum tanker reposition time $\max[ttrv_{ar}]$ specifies m. The elastic demand constraint delivery windows are typically between 1 and 10 days.

Because of the cascade convention stipulating that no column's time index exceed an associated row's time index, we re-define the $GTONS_{ict}$ variable. For the cascade formulation, $GTONS_{ict}$ is the amount of i's cargo of type c **transshipped** on day t, but only when $t + gtrv_i \in TW_i$. Since $GTONS_{ict}$ appears only in the objective function, the $MEETDEM$ constraint, and the $TRANSTONS$ constraint (and is effectively constrained only by the latter), the change has minimal impact on the formulation. This adjustment also has the advantage of reducing the number of staircase rows, since each $TRANSTONS$ constraint includes columns from only one time period.

In addition to the notation defined previously, let $fix(ROW^n)$ represent all terms in equation ROW that were fixed prior to subproblem n. Stated another way, $fix(ROW^n)$ is the sum of all associated fixed columns, *i.e.*, those that are indexed by $t \in \bigcup_{n'<n} TF^{n'}$.

For each $n \in NC^n$, the nth subproblem formulation follows

OBJ^n: Objective function

minimize

$$\sum_{i\in I}\sum_{a\in A}\sum_{c\in C_a,\, t\in TW_i\cap TC^n} latepen_i \cdot (t - rdd_i)^+ \cdot DTONS_{iact}$$

$$+ \sum_{i\in I_{fob}}\sum_{a\in A}\sum_{c\in C_a}\sum_{t\in TW_i\cap TC^n} latepen_i \cdot (t - rdd_i)^+ \cdot STONS_{iact}$$

$$+ \sum_{i\in I_{fob}}\sum_{c\in C}\sum_{t\in TC^n} latepen_i \cdot (t + gtrv_i - rdd_i)^+ \cdot GTONS_{ict}$$

$$+ \sum_{i\in I}\sum_{c\in C} nogopen_i \cdot NOGO_{ic} \cdot I\left(T_i \cap TC^n \neq \emptyset\right)$$

$$+ \sum_{a\in A_{chp}}\sum_{b\in B_{sup}}\sum_{t\in TC^n} usepen_a \cdot [THCHOP_{abt} + THCHOPR_{abt}]$$

$$+ \sum_{a\in A_{tkr}}\sum_{b\in B_e}\sum_{t\in TC^n} usepen_a \cdot TKREC_{abt} + \sum_{a\in A_{tkr}}\sum_{b\in B_{tkr}}\sum_{t\in TC^n} usepen_a \cdot TKRBC_{abt}$$

$$- \sum_{a\in A}\sum_{b\in B_e}\sum_{t\in TC^n} restrew_a \cdot RON_{abt} + \sum_{a\in A}\sum_{b,b'\in B_{crw}}\sum_{t\in TC^n} dhpen_a \cdot DHCREW_{abb't}$$

$$+ \sum_{i\in I}\sum_{n'<n} nogopen_i \cdot NOGO_{ic}^{n'} \cdot I\left(lastp^{n'-1} < rdd_i + maxlate_i \leq lastp^{n'}\right)$$

$$+ fix(obj^n)$$

This objective is similar to the monolith objective, with the following exceptions. Column indexed by time are active only if the time index is active. Since $NOGO_{ic}$ is null indexed (not indexed by time), it is active only if a period of the line id's delivery window is active. The late penalty on $GTONS_{ict}$ is adjusted for that variable's re-definition.

The objective also includes two constants: 1) $\sum_{i\in I}\sum_{n'<n} nogopen_i \cdot NOGO_{ic}^{n'}$, the accumulated non-delivery penalties from line id's whose delivery window is now a subset of fixed periods; and 2) $fix(obj^n)$, the accumulated penalties from previous subproblems' columns indexed by t.

ACBALE^n: aircraft balance at embarkation nodes

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XT_{iart} + \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XD_{iart}$$

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XTR_{iart} + \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XDR_{iart}$$

$$+I(a \in A_{tkr}) \cdot [TKREC_{abt}] + RON_{abt}$$

$$= RON_{abt-1} \cdot I(t-1 \in TC^n) + \sum_{r \in RB_{ab}} Y_{art-trv_{ar}} \cdot I(t - trv_{ar} \in TC^n)$$

$$+ALLOC_{abt} + I(a \in A_{tkr}) \cdot [TKRCE_{abt}] + fix(ACBALE^n_{abt})$$

$$\forall a \in A, b \in B_e, t \in TC^n$$

The cascade modification adds a fixed term, and shows only active rows and columns: those indexed by an active period.

ACBALSUP^n: aircraft balance at SUPER debarkation nodes

$$\sum_{r \in RB_{ab} \cap \bar{R}\bar{B}_{rec}} Y_{art} + RONT_{abt} + THCHOP_{abt} =$$

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XT_{iart-trv_{ar}} \cdot I(t - trv_{ar} \in TC^n)$$

$$+ \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XD_{iart-trv_{ar}} \cdot I(t - trv_{ar} \in TC^n)$$

$$+(RONT_{abt-1} + THCHOP_{abt-1}) \cdot I(t-1 \in TC^n)$$

$$+I(t=1) \cdot IRONT_{ab} + fix(ACBALSUP^n_{abt}) \qquad \forall a \in A, b \in B_{sup}, t \in TC^n$$

The cascade modification adds a fixed term, and shows only active rows and columns: those indexed by an active period, and $IRONT_{ab}$ for $t = 1$.

ACBALREC^n: aircraft balance at SUPER debarkation nodes with recovery

$$\sum_{r \in RB_{ab} \cap RB_{rec}} Y_{art} + RONR_{abt} + THCHOPR_{abt} =$$
$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XTR_{iart-trv_{ar}} \cdot I(t - trv_{ar} \in TC^n)$$
$$+ \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XDR_{iart-trv_{ar}} \cdot I(t - trv_{ar} \in TC^n)$$
$$+ (RONR_{abt-1} + THCHOPR_{abt-1}) \cdot I(t - 1 \in TC^n)$$
$$+ I(t = 1) \cdot IRONR_{ab} + fix(ACBALREC^n_{abt}) \qquad \forall a \in A, b \in BS_{rec}, t \in TC^n$$

The cascade modification adds a fixed term, and shows only active rows and columns: those indexed by an active period, and $IRONR_{ab}$ for $t = 1$.

INITIRON^n: allocate initial chops to recovery or not

$$IRONT_{ab} + IRONR_{ab} = initchop_{ab} \quad \forall a \in A_{chp}, b \in B_{sup}, n = 1$$

The cascade modification activates rows and columns of this constraint only in the first subproblem.

ACALLOC^n: allocate newly available aircraft

$$\sum_{b \in B_e} ALLOC_{abt} = newac_{at} \quad \forall a \in A, t \in TC^n$$

The cascade modification shows rows and columns are active only if indexed by an active period.

SHUTLBND^n: don't send more shuttles than available

$$\sum_{i \in I_{b,sup} \cap I_{fob}} \frac{XS_{iat}}{shutrate_{ai}} \leq [THCHOP_{abt} + THCHOPR_{abt}] \quad \forall a \in A, b \in B_{sup}, t \in TC^n$$

The cascade modification shows rows and columns are active only if indexed by an active period.

TKRBND^n: don't use more tankers than available

$$\sum_{b' \in BA_{b,tkr}} \frac{TKRA_{abb't}}{tkrrate_{abb'}} \leq TKRB_{abt} \quad \forall a \in A_{tkr}, b \in B_{tkr}, t \in TC^n$$

The cascade modification shows rows and columns are active only if indexed by an active period.

CLOUDBAL^n: flow balance: leaving and entering tanker fleet

$$\sum_{b \in B_e} TKREC_{abt-ttrv_{ab}} \cdot I(t - ttrv_{ab} \in TC^n)$$
$$+ \sum_{b \in B_{tkr}} TKRBC_{abt-ttrv_{ab}}) \cdot I(t - ttrv_{ab} \in TC^n)$$
$$+ fix(CLOUDBAL^n_{at}) =$$
$$\sum_{b \in B_e} TKRCE_{abt} + \sum_{b \in B_{tkr}} TKRCB_{abt} \qquad \forall a \in A_{tkr}, t \in TC^n$$

The cascade modification adds a fixed term, and shows rows and columns are active only if indexed by an active period.

TKRINVT^n: tanker inventory at tanker bed-downs

$$TKRBC_{abt} + TKRB_{abt} = TKRCB_{abt} +$$
$$TKRB_{abt-1} \cdot I(t - 1 \in TC^n) + fix(TKRINVT^n_{abt}) \quad \forall a \in A_{tkr}, b \in B_{tkr}, t \in TC^n$$

The cascade modification adds a fixed term, and shows rows and columns are active only if indexed by an active period.

ARMOG^n: aerial refueling capacity constraint

$$\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_b\cap RD_{ia,dir}} tkreqvs_{abr}\cdot XD_{iart-etrv_{abr}}\cdot I(t-etrv_{abr}\in TC^n)$$

$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_b\cap RD_{ia,trn}} tkreqvs_{abr}\cdot XT_{iart-etrv_{abr}}\cdot I(t-etrv_{abr}\in TC^n)$$

$$\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_b\cap RD_{ia,dir}} tkreqvs_{abr}\cdot XDR_{iart-etrv_{abr}}\cdot I(t-etrv_{abr}\in TC^n)$$

$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_b\cap RD_{ia,trn}} tkreqvs_{abr}\cdot XTR_{iart-etrv_{abr}}\cdot I(t-etrv_{abr}\in TC^n)$$

$$+\sum_{a\in A_{rfl}}\sum_{r\in RB_{ab}} tkreqvs_{abr}\cdot Y_{art-etrv_{abr}}\cdot I(t-etrv_{abr}\in TC^n)$$

$$+fix(ARMOG^n_{bt})$$

$$\leq \sum_{b'\in BT_{b,arp}}\sum_{a\in A_{tkr}} tkrprop_{ab'b}\cdot TKRA_{ab'bt} \qquad \forall b\in B_{arp}, t\in TC^n$$

The cascade modification adds a fixed term, and shows rows and columns are active only if indexed by an active period.

UTE^n: utilization rate

$$\sum_{t\in T_u}\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} flttime_{arf}\cdot XD_{iart-(f-1)}\cdot I\left(t-(f-1)\in TC^n\right)$$
$$+\sum_{t\in T_u}\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} flttime_{arf}\cdot XT_{iart-(f-1)}\cdot I\left(t-(f-1)\in TC^n\right)$$
$$+\sum_{t\in T_u}\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} flttime_{arf}\cdot XDR_{iart-(f-1)}\cdot I\left(t-(f-1)\in TC^n\right)$$
$$+\sum_{t\in T_u}\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} flttime_{arf}\cdot XTR_{iart-(f-1)}\cdot I\left(t-(f-1)\in TC^n\right)$$
$$+\sum_{i\in I_{fob}}\sum_{t\in T_u} shuttime_{ia}\cdot XS_{iat}$$
$$+\sum_{t\in T_u}\sum_{r\in RB_b}\sum_{f\in FT} flttime_{arf}\cdot Y_{art-(f-1)}\cdot I\left(t-(f-1)\in TC^n\right)$$
$$+I(a\in A_{tkr})\cdot\left[\sum_{b\in B_{tkr}}\sum_{b'\in B_{arp}}\sum_{t\in T_u} tkrtime_{abb'}\cdot TKRA_{abb't}\right.$$
$$+\sum_{b\in B_e}\sum_{t\in T_u} hrsper\cdot rttrv_{ab}\cdot TKREC_{abt}$$
$$\left.+\sum_{b\in B_{tkr}}\sum_{t\in T_u} hrsper\cdot rttrv_{ab}\cdot TKRBC_{abt}\right]+fix(UTE^n_{au})$$
$$\leq\sum_{t\in T_u} cumac_{at}\cdot urate_a \qquad \forall a\in A,\ \ \forall u: T_u\cap TC^n\neq\emptyset$$

The cascade modification adds a fixed term, shows columns are active only if indexed by an active period, and show rows are active only if a time period in the ute rate block u is active. Although this constraint is much wider than the cascade overlap, feasibility is not jeopardized because of the constraint's sense. However, it may jeopardize proximal cascade quality if tight in the solution to the monolith.

ACCONSUME^n: max acft usage to lessen rounding effects

$$\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} msntime_{arf}\cdot XD_{iart-(f-1)}\cdot I\,(t-(f-1)\in TC^n)$$

$$+\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} msntime_{arf}\cdot XT_{iart-(f-1)}\cdot I\,(t-(f-1)\in TC^n)$$

$$\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} msntime_{arf}\cdot XDR_{iart-(f-1)}\cdot I\,(t-(f-1)\in TC^n)$$

$$+\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} msntime_{arf}\cdot XTR_{iart-(f-1)}\cdot I\,(t-(f-1)\in TC^n)$$

$$+\sum_{i\in I_{fob}}\frac{hrsper}{shutrate_{ai}}\cdot XS_{iat}$$

$$+\sum_{r\in RB}\sum_{f\in FT} msntime_{arf}\cdot Y_{art-(f-1)}\cdot I\,(t-(f-1)\in TC^n)$$

$$+I(a\in A_{tkr})\cdot\left[\sum_{b\in B_{tkr}}\sum_{b'\in B_{arp}}\frac{hrsper}{tkrrate_{abb'}}\cdot TKRA_{abb't}\right.$$

$$+\sum_{b\in B_e} rttrv_{ab}\cdot hrsper\cdot TKREC_{abt}$$

$$\left.+\sum_{b\in B_{tkr}} rttrv_{ab}\cdot hrsper\cdot TKRBC_{abt}\right]$$

$$+\sum_{b\in B_e} hrsper\cdot RON_{abt}+\sum_{b\in B_{sup}} hrsper\cdot[RONT_{abt}+RONR_{abt}]$$

$$+fix(ACCONSUME^n_{at})$$

$$\leq hrsper\cdot cumac_{at} \qquad \forall a\in A, t\in TC^n$$

The cascade modification adds a fixed term, and shows rows and columns are active only if indexed by an active period.

DCAPACITY^n: aircraft capacity for direct delivery

$$\sum_{c \in C_a \cap CC} \frac{DTONS_{iact}}{purecap_{iac}} + \frac{paxfrac_a \cdot DTONS_{i,a,pax,t}}{maxpax_a} \cdot I(a \in A_{pax}) \leq$$

$$\sum_{r \in RD_{ia,dir}} rangefac_{iar} \cdot [XD_{iart-trv_{ar}} + XDR_{iart-trv_{ar}}] \cdot I(t - trv_{ar} \in TC^n)$$

$$+fix(DCAPACITY^n_{iat}) \qquad \forall i \in I, a \in A, t \in TC^n \cap TW_i$$

The cascade modification adds a fixed term, and shows rows and columns are active only if indexed by an active period.

TCAPACITY^n: aircraft capacity for transshipments

$$\sum_{c \in C_a \cap CC} \frac{TTONS_{iact}}{purecap_{iac}} + \frac{paxfrac_a \cdot TTONS_{i,a,pax,t}}{maxpax_a} \cdot I(a \in A_{pax}) \leq$$

$$\sum_{r \in RD_{ia,trn}} rangefac_{iar} \cdot [XT_{iart-trv_{ar}} + XTR_{iart-trv_{ar}}] \cdot I(t - trv_{ar} \in TC^n)$$

$$+fix(TCAPACITY^n_{iat}) \qquad \forall i \in I_{fob}, a \in A, t \in TC^n \cap TW_i$$

The cascade modification adds a fixed term, and shows rows and columns are active only if indexed by an active period.

SCAPACITY^n: aircraft capacity for shuttle deliveries

$$\sum_{c \in C_a \cap CC} \frac{STONS_{iact}}{purecap_{iac}} + \frac{paxfrac_a \cdot STONS_{i,a,pax,t}}{maxpax_a} \cdot I(a \in A_{pax})$$

$$\leq srange_{ia} \cdot XS_{iat} \qquad i \in I_{fob}, a \in A, t \in TC^n \cap TW_i$$

The cascade modification shows rows and columns are active only if indexed by an active period.

DPAXCAP^n: aircraft capacity for direct delivery of pax

$$DTONS_{i,a,pax,t} \leq$$
$$\sum_{r \in RD_{ia,dir}} maxpax_a \cdot [XD_{iart-trv_{ar}} + XDR_{iart-trv_{ar}}] \cdot I(t - trv_{ar} \in TC^n)$$
$$+fix(DPAXCAP^n_{iat}) \qquad \forall i \in I, a \in A_{mix}, t \in TC^n \cap TW_i$$

The cascade modification adds a fixed term, and shows rows and columns are active only if indexed by an active period.

TPAXCAP^n: aircraft capacity for transshipment of pax

$$TTONS_{i,a,pax,t} \leq$$
$$\sum_{r \in RD_{ia,trn}} maxpax_a \cdot [XT_{iart-trv_{ar}} + XTR_{iart-trv_{ar}}] \cdot I(t - trv_{ar} \in TC^n)$$
$$+fix(TPAXCAP^n_{iat}) \qquad \forall i \in I_{fob}, a \in A_{mix}, t \in TC^n \cap TW_i$$

The cascade modification adds a fixed term, and shows rows and columns are active only if indexed by an active period.

SPAXCAP^n: aircraft capacity for delivery of pax by shuttles

$$STONS_{i,a,pax,t} \leq maxpax_a \cdot XS_{iat} \qquad \forall i \in I_{fob}, a \in A_{mix}, t \in TC^n \cap TW_i$$

The cascade modification shows rows and columns are active only if indexed by an active period.

MEETDEM^n: meet demand for each line id

$$\sum_{a\in A_c}\sum_{t\in TC^n\cap TW_i} DTONS_{iact} + NOGO_{ic} + fix(MEETDEM^n_{ic})$$
$$+I(i\in I_{fob})\cdot\left[\sum_{a\in A_c}\sum_{t\in TC^n\cap TW_i} STONS_{iact} + \sum_{t\in TC^n} GTONS_{ict}\right] = dem_{ic}$$
$$\forall c\in C,\quad \forall i : TW_i\cap TC^n\neq\emptyset$$

The cascade modification includes a term for fixed deliveries, shows columns are active only if indexed by an active period, and shows rows are active only if the delivery window includes an active period. Note also that $GTONS_{ict}$ is defined only for $t+gtrv_i \in TW_i$.

TRANSTONS^n: flow balance for transshipped stons

$$\sum_{a\in A_c} TTONS_{iact} = \sum_{a\in A_c} STONS_{iact} + GTONS_{ict} \quad \forall i\in I_{fob}, c\in C, t\in TW_i\cap TC^n$$

Because of the time-index shift in the $GTONS_{ict}$ re-definition, this constraint no longer links multiple time periods. Since $GTONS_{ict}$ is restricted to $t+gtrv_i\in TW_i$, the only explicit cascade modifications show columns are active only if indexed by an active period, and rows are active only if a period of the line id's delivery window is active.

INITCREWS$^{n=1}$: initialize crew placement

$$\sum_{b\in B_{crw}} SCREWS_{abt} + crewrat_a\cdot\sum_{b\in B_{tkr}} TKRB_{abt}$$
$$= crewrat_a\cdot newac_{at} \qquad \forall a, t=1$$

The cascade modification shows rows and columns are only active for the first time period.

SCREWBAL^n: strategic crew balance of flow

$$SCREWS_{abt+1} = SCREWS_{abt} \cdot I(t \in TC^n)$$
$$+\sum_{i \in I} \sum_{r \in RD_{ia,dir} \cap \bar{R}_{b,ori}} [XD_{iart-ctrv_{abr}} + XDR_{iart-ctrv_{abr}}] \cdot I(t - ctrv_{abr} \in TC^n)$$
$$+\sum_{i \in I_{fob}} \sum_{r \in RD_{ia,trn} \cap \bar{R}_{b,ori}} [XD_{iart-ctrv_{abr}} + XDR_{iart-ctrv_{abr}}] \cdot I(t - ctrv_{abr} \in TC^n)$$
$$+ \sum_{r \in RB \cap \bar{R}_{b,ori}} Y_{art-ctrv_{abr}} \cdot I(t - ctrv_{abr} \in TC^n)$$
$$-\sum_{i \in I} \sum_{r \in RD_{ia,dir} \cap \bar{R}_{b,dst}} [XD_{iart-etrv_{abr}} + XDR_{iart-etrv_{abr}}] \cdot I(t - etrv_{abr} \in TC^n)$$
$$-\sum_{i \in I_{fob}} \sum_{r \in RD_{ia,trn} \cap \bar{R}_{b,dst}} [XT_{iart-etrv_{abr}} + XTR_{iart-etrv_{abr}}] \cdot I(t - etrv_{abr} \in TC^n)$$
$$- \sum_{r \in RB \cap \bar{R}_{b,dst}} Y_{art-etrv_{abr}} \cdot I(t - etrv_{abr} \in TC^n)$$
$$+I(b \in B_e, t - cttrv_{ab} \in TC^n) \cdot crewrat_a \cdot TKRCE_{abt-cttrv_{ab}}$$
$$-I(b \in B_e, t \in TC^n) \cdot crewrat_a \cdot TKREC_{abt}$$
$$+I(b \in B_{sup}, t-1 \in TC^n) \cdot crewrat_a \cdot THCHOP_{abt-1}$$
$$-I(b \in B_{sup}, t \in TC^n) \cdot crewrat_a \cdot THCHOP_{abt} + I(1 \in TC^n) \cdot IRONT_{a,b}$$
$$+I(b \in BS_{rec}, t-1 \in TC^n) \cdot crewrat_a \cdot THCHOPR_{abt-1}$$
$$-I(b \in BS_{rec}, t \in TC^n) \cdot crewrat_a \cdot THCHOPR_{abt} + I(1 \in TC^n) \cdot IRONR_{a,b}$$
$$+I(b \in B_e, t \neq 1, newac_{at} > 0, t \in TC^n) \cdot crewrat_a \cdot ALLOC_{abt}$$
$$+ \sum_{b' \in B_{crw}} DHCREW_{ab'bt-dhtrv_{b'b}} \cdot I(t - dhtrv_{b'b} \in TC^n)$$
$$- \sum_{b' \in B_{crw}} DHCREW_{abb't} \cdot I(t \in TC^n) + fix(SCREWBAL^n_{abt+1})$$
$$\forall a \in A, b \in B_{crw}, \ \forall t : (t \in T, t+1 \in TC^n)$$

In order to satisfy the cascade requirement that no column have a time index greater than an associated row's time index, we define this constraint on $t+1$. The cascade modification also adds a fixed term, and shows only active rows and columns: those indexed by an active period, and $IRONT_{ab}$ and $IRONR_{ab}$ for $t = 1$.

MOG^n: airfield capacity

$$\sum_{i\in I}\sum_{a\in A}\sum_{r\in RD_b\cap RD_{ia,dir}} gtime_{abr}\cdot acpkg_{ab}\cdot[XD_{iart-etrv_{abr}}+XDR_{iart-etrv_{abr}}]$$
$$\cdot I(t-etrv_{abr}\in TC^n)$$
$$+\sum_{i\in I_{b,dst}}\sum_{a\in A}\sum_{r\in RD_{ia,dir}} gtime_{abr}\cdot acpkg_{ab}\cdot XD_{iart-trv_{ar}}\cdot I(t-trv_{ar}\in TC^n)$$
$$+\sum_{i\in I_{b,dst}}\sum_{a\in A}\sum_{r\in RD_{ia,dir}} qtime_{abr}\cdot acpkg_{ab}\cdot XDR_{iart-trv_{ar}}\cdot I(t-trv_{ar}\in TC^n)$$
$$+\sum_{i\in I}\sum_{a\in A}\sum_{r\in RD_b\cap RD_{ia,trn}} gtime_{abr}\cdot acpkg_{ab}\cdot[XT_{iart-etrv_{abr}}+XTR_{iart-etrv_{abr}}]$$
$$\cdot I(t-etrv_{abr}\in TC^n)$$
$$+\sum_{i\in I_{b,trn}}\sum_{a\in A}\sum_{r\in RD_{ia,trn}} gtime_{abr}\cdot acpkg_{ab}\cdot XT_{iart-trv_{ar}}\cdot I(t-trv_{ar}\in TC^n)$$
$$+\sum_{i\in I_{b,trn}}\sum_{a\in A}\sum_{r\in RD_{ia,trn}} qtime_{abr}\cdot acpkg_{ab}\cdot XTR_{iart-trv_{ar}}\cdot I(t-trv_{ar}\in TC^n)$$
$$+\sum_{i\in(I_{b,dst}\cap I_{fob})\cup I_{b,trn}}\sum_{a\in A} sgtime_{ab}\cdot acpkg_{ab}\cdot XS_{iat}$$
$$+\sum_{b'\in B_{sup}\cap BS_{b,dwn}}\sum_{a\in A} hrsper\cdot acpkg_{ab}\cdot[THCHOP_{ab't}+THCHOPR_{ab't}]$$
$$+\sum_{a\in A}\sum_{r\in RB_{ab}} gtime_{abr}\cdot acpkg_{ab}\cdot Y_{art-etrv_{abr}}\cdot I(t-etrv_{abr}\in TC^n)$$
$$+I(b\in B_{tkr})\cdot\left[\sum_{a\in A_{tkr}} hrsper\cdot acpkg_{ab}\cdot TKRB_{abt}\right]$$
$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,dir}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XD_{iart-etrv_{abr}}$$
$$\cdot I(t-etrv_{abr}\in TC^n)$$
$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,dir}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XDR_{iart-etrv_{abr}}$$
$$\cdot I(t-etrv_{abr}\in TC^n)$$
$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,trn}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XT_{iart-etrv_{abr}}$$
$$\cdot I(t-etrv_{abr}\in TC^n)$$
$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,trn}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XTR_{iart-etrv_{abr}}$$
$$\cdot I(t-etrv_{abr}\in TC^n)$$
$$+\sum_{a\in A_{rfl}}\sum_{r\in RB_{b,div}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot Y_{art-etrv_{abr}}\cdot I(t-etrv_{abr}\in TC^n)$$
$$+fix(MOG^n_{bt})$$
$$\leq mog_b\cdot mogeff_b \qquad \forall b\in B\setminus B_{sup}\setminus B_{arp}\setminus B_{way}, t\in TC^n$$

The cascade modification adds a fixed term, and shows rows and columns are active only if indexed by an active period.

Non-negativity of all variables.

D. NRMO BY LAGRANGIAN CASCADE

With the exception of the objective function, the Lagrangian cascade formulation of NRMO is straightforward. Rows indexed by $t \in TRL^{\ell}$ are active in subproblem ℓ. Additionally, rows indexed by $t \in TO^{\ell}$ are active if all technological coefficients are positive, and the row has sense "$\leq$". Columns are active in subproblem ℓ if indexed by $t \in TRL^{\ell} \cup TO^{\ell}$, the extended-active set.

In addition to the notation used in Chapter II, define the following:

ald_i	Available to load date, the first period in TW_i
RX	The set of rows that link two or more sets TRL^{ℓ}. This is the Lagrange-relaxed constraint set.
T_u^{ℓ}	Ute rate block defined as the active set, $T_u^{\ell} \equiv TRL^{\ell}$

Additionally, any variable in the formulation indexed by $t \in TO^{\ell}$ (the extended set) is a duplicate variable; one that is unique to subproblem ℓ.

For each $\ell \in CL$:

OBJ$^{\ell}$: Objective function

minimize

$$\sum_{i\in I}\sum_{a\in A}\sum_{c\in C_a,\, t\in TW_i\cap TRL^{\ell}} latepen_i \cdot (t - rdd_i)^{+} \cdot DTONS_{iact}$$

$$+ \sum_{i\in I_{fob}}\sum_{a\in A}\sum_{c\in C_a}\sum_{t\in TW_i\cap TRL^{\ell}} latepen_i \cdot (t - rdd_i)^{+} \cdot STONS_{iact}$$

$$+ \sum_{i\in I_{fob}}\sum_{c\in C}\sum_{t\in TRL^{\ell}} latepen_i \cdot (t + gtrv_i - rdd_i)^{+} \cdot GTONS_{ict}$$

$$+ \sum_{i\in I}\sum_{c\in C} nogopen_i \cdot NOGO_{ic} \cdot I\left(i : TW_i \subseteq TRL^{\ell}\right)$$

$$+ \sum_{a\in A_{chp}}\sum_{b\in B_{sup}}\sum_{t\in T\cap TRL^{\ell}} usepen_a \cdot [THCHOP_{abt} + THCHOPR_{abt}]$$

$$+ \sum_{a\in A_{tkr}}\sum_{b\in B_e}\sum_{t\in T\cap TRL^{\ell}} usepen_a \cdot TKREC_{abt}$$

$$+ \sum_{a\in A_{tkr}}\sum_{b\in B_{tkr}}\sum_{t\in T\cap TRL^{\ell}} usepen_a \cdot TKRBC_{abt}$$

$$- \sum_{a\in A}\sum_{b\in B_e}\sum_{t\in T\cap TRL^{\ell}} restrew_a \cdot RON_{abt}$$

$$+ \sum_{a\in A}\sum_{b,b'\in B_{crw}}\sum_{t\in T\cap TRL^{\ell}} dhpen_a \cdot DHCREW_{abb't}$$

$$+ \ell term$$

The objective function is similar to the monolith's objective function, but columns are active only if indexed by an active period, $t \in TRL^{\ell}$. It includes the $NOGO_{ic}$ columns for a line id only if i's delivery window is a subset of the active periods. The definition of $GTONS_{ict}$ is identical to the one used in the proximal cascade formulation.

The objective function also includes $\ell term$, which is the Lagrange penalty term for all active columns associated with Lagrange-relaxed rows. The coefficient on each of the Lagrange penalty terms in the objective is quite complex. Rather than list them entirely, we list the coefficient on the variable XT_{iart} only. Dual multipliers β corresponding to

Lagrange-relaxed rows are superscripted with the row's name:

$$\sum_{i\in I_{fob}}\sum_{a\in A}\sum_{r\in RD_b\cap RD_{ia,trn}}\sum_{t\in TRL^{\ell}} XT_{iart}\cdot\left[\sum_{b\in B_e}\beta_{abt}^{ACBALE}\cdot I\left(ACBALE_{abt}\in RX\right)\right.$$

$$+\sum_{b\in B_{sup}}\beta_{abt+trv_{ar}}^{ACBALSUP}\cdot I\left(ACBALSUP_{abt}\in RX\right)$$

$$-\sum_{b\in B_{arp}}\beta_{abt+etrv_{abr}}^{ARMOG}\cdot tkreqvs_{abr}\cdot I\left(ARMOG_{abt+etrv_{abr}}\in RX\right)$$

$$-\sum_{f\in FT}\beta_{at+(f-1)}^{UTE}\cdot flttime_{arf}\cdot I\left(t+(f-1)\in TRL^{\ell+1}\right)$$

$$-\sum_{f\in FT}\beta_{at+(f-1)}^{ACCONSUME}\cdot msntime_{arf}\cdot I\left(ACCONSUME_{art+(f-1)}\in RX\right)$$

$$+\beta_{iat+trv_{ar}}^{TCAPACITY}\cdot rangefac_{iar}\cdot I\left(TCAPACITY_{iat+trv_{ar}}\in RX\right)$$

$$+\beta_{iat+trv_{ar}}^{TPAXCAP}\cdot \max pax_{a}\cdot I\left(TPAXCAP_{iat+trv_{ar}}\in RX\right)$$

$$-\sum_{b\in B_{crw}}\beta_{abt+1+ctrv_{abr}}^{SCREWBAL}\cdot I\left(SCREWBAL_{abt+1+ctrv_{abr}}\in RX\right)$$

$$+\sum_{b\in B_{crw}}\beta_{abt+1+etrv_{abr}}^{SCREWBAL}\cdot I\left(SCREWBAL_{abt+1+etrv_{abr}}\in RX\right)$$

$$-\sum_{b\in B\setminus B_{\sup}\setminus B_{arp}\setminus B_{way}}\beta_{bt+etrv_{abr}}^{MOG}\cdot gtime_{abr}\cdot acpkg_{ab}\cdot I\left(MOG_{bt+etrv_{abr}}\in RX\right)$$

$$-\sum_{b\in B\setminus B_{\sup}\setminus B_{arp}\setminus B_{way}}\beta_{bt+trv_{ar}}^{MOG}\cdot gtime_{abr}\cdot acpkg_{ab}\cdot I\left(MOG_{bt+trv_{ar}}\in RX\right)$$

$$\left.-\sum_{b\in B\setminus B_{\sup}\setminus B_{arp}\setminus B_{way}}\beta_{bt+trv_{ar}}^{MOG}\cdot dpct_{a}\cdot gtime_{abr}\cdot acpkg_{ab}\cdot I\left(MOG_{bt+trv_{ar}}\in RX\right)\right]$$

The Lagrangian cascade solution value is given by the sum of the subproblem objective values, plus the sum of the Lagrange-relaxed row right-hand-sides multiplied by their associated penalties.

ACBALE$^{\ell}$: aircraft balance at embarkation nodes

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XT_{iart} + \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XD_{iart}$$

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XTR_{iart} + \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XDR_{iart}$$

$$+I(a \in A_{tkr}) \cdot [TKREC_{abt}] + RON_{abt} = RON_{abt-1} + \sum_{r \in RB_{ab}} Y_{art-trv_{ar}}$$

$$+ALLOC_{abt} + I(a \in A_{tkr}) \cdot [TKRCE_{abt}] \qquad \forall a \in A, b \in B_e, t \in TRL^{\ell}$$

The cascade modification activates rows and columns indexed by the active set, and duplicates columns indexed by the extended set.

ACBALSUP$^{\ell}$: aircraft balance at SUPER debarkation nodes

$$\sum_{r \in RB_{ab} \cap \bar{R}\bar{B}_{rec}} Y_{art} + RONT_{abt} + THCHOP_{abt} =$$

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XT_{iart-trv_{ar}} + \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XD_{iart-trv_{ar}}$$

$$+RONT_{abt-1} + THCHOP_{abt-1} + I(t = 1) \cdot IRONT_{ab} \qquad \forall a \in A, b \in B_{sup}, t \in TRL^{\ell}$$

The cascade modification activates rows and columns indexed by the active set, and duplicates columns indexed by the extended set. $IRONT_{ab}$ is only active in the first subproblem.

ACBALREC$^{\ell}$: aircraft balance at SUPER debarkation nodes with recovery

$$\sum_{r \in RB_{ab} \cap RB_{rec}} Y_{art} + RONR_{abt} + THCHOPR_{abt} =$$

$$\sum_{i \in I_{fob}} \sum_{r \in RD_b \cap RD_{ia,trn}} XTR_{iart-trv_{ar}} + \sum_{i \in I} \sum_{r \in RD_b \cap RD_{ia,dir}} XDR_{iart-trv_{ar}}$$

$$+RONR_{abt-1} + THCHOPR_{abt-1} + I(t = 1) \cdot IRONR_{ab} \quad \forall a \in A, b \in BS_{rec}, t \in TRL^{\ell}$$

The cascade modification activates rows and columns indexed by the active set,

and duplicates columns indexed by the extended set. $IRONR_{ab}$ is only active in the first subproblem.

INITIRON$^{\ell=1}$: allocate initial chops to recovery or not

$$IRONT_{ab} + IRONR_{ab} = initchop_{ab} \quad \forall a \in A_{chp}, b \in B_{sup}$$

The cascade modification activates these rows and columns only in the first subproblem.

ACALLOC$^{\ell}$: allocate newly available aircraft

$$\sum_{b \in B_e} ALLOC_{abt} = newac_{at} \quad \forall a \in A, t \in TRL^{\ell}$$

The cascade modification activates rows and columns indexed by the active set.

SHUTLBND$^{\ell}$: don't send more shuttles than available

$$\sum_{i \in I_{b,sup} \cap I_{fob}} \frac{XS_{iat}}{shutrate_{ai}} \leq [THCHOP_{abt} + THCHOPR_{abt}] \quad \forall a \in A, b \in B_{sup}, t \in TRL^{\ell}$$

The cascade modification activates rows and columns indexed by the active set.

TKRBND$^{\ell}$: don't use more tankers than available

$$\sum_{b' \in BA_{b,tkr}} \frac{TKRA_{abb't}}{tkrrate_{abb'}} \leq TKRB_{abt} \quad \forall a \in A_{tkr}, b \in B_{tkr}, t \in TRL^{\ell}$$

The cascade modification activates rows and columns indexed by the active set.

CLOUDBAL$^{\ell}$: flow balance: leaving and entering tanker fleet

$$\sum_{b \in B_e} TKREC_{abt-ttrv_{ab}} + \sum_{b \in B_{tkr}} TKRBC_{abt-ttrv_{ab}} =$$

$$\sum_{b \in B_e} TKRCE_{abt} + \sum_{b \in B_{tkr}} TKRCB_{abt} \qquad \forall a \in A_{tkr}, t \in TRL^{\ell}$$

The cascade modification activates rows and columns indexed by the active set, and duplicates columns indexed by the extended set.

TKRINVT$^{\ell}$: tanker inventory at tanker bed-downs

$$TKRBC_{abt} + TKRB_{abt} = TKRCB_{abt} + TKRB_{abt-1} \quad \forall a \in A_{tkr}, b \in B_{tkr}, t \in TRL^{\ell}$$

The cascade modification activates rows and columns indexed by the active set, and duplicates columns indexed by the extended set.

ARMOG$^{\ell}$: aerial refueling capacity constraint

$$\sum_{i \in I} \sum_{a \in A_{rfl}} \sum_{r \in RD_b \cap RD_{ia,dir}} tkreqvs_{abr} \cdot XD_{iart-etrv_{abr}} \cdot I(t - etrv_{abr} \in TRL^{\ell} \cup TO^{\ell})$$

$$+ \sum_{i \in I} \sum_{a \in A_{rfl}} \sum_{r \in RD_b \cap RD_{ia,trn}} tkreqvs_{abr} \cdot XT_{iart-etrv_{abr}} \cdot I(t - etrv_{abr} \in TRL^{\ell} \cup TO^{\ell})$$

$$\sum_{i \in I} \sum_{a \in A_{rfl}} \sum_{r \in RD_b \cap RD_{ia,dir}} tkreqvs_{abr} \cdot XDR_{iart-etrv_{abr}} \cdot I(t - etrv_{abr} \in TRL^{\ell} \cup TO^{\ell})$$

$$+ \sum_{i \in I} \sum_{a \in A_{rfl}} \sum_{r \in RD_b \cap RD_{ia,trn}} tkreqvs_{abr} \cdot XTR_{iart-etrv_{abr}} \cdot I(t - etrv_{abr} \in TRL^{\ell} \cup TO^{\ell})$$

$$+ \sum_{a \in A_{rfl}} \sum_{r \in RB_{ab}} tkreqvs_{abr} \cdot Y_{art-etrv_{abr}} \cdot I(t - etrv_{abr} \in TRL^{\ell} \cup TO^{\ell})$$

$$\leq \sum_{b' \in BT_{b,arp}} \sum_{a \in A_{tkr}} tkrprop_{ab'b} \cdot TKRA_{ab'bt} \qquad \forall b \in B_{arp}, t \in TRL^{\ell} \cup TO^{\ell}$$

The cascade modification activates rows indexed by the extended-active set, columns indexed by the active set and duplicates columns indexed by the extended set.

UTE$^{\ell}$: utilization rate

$$\sum_{t\in T_u^\ell}\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} flttime_{arf}\cdot XD_{iart-(f-1)}$$
$$+\sum_{t\in T_u^\ell}\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} flttime_{arf}\cdot XT_{iart-(f-1)}$$
$$+\sum_{t\in T_u^\ell}\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} flttime_{arf}\cdot XDR_{iart-(f-1)}$$
$$+\sum_{t\in T_u^\ell}\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} flttime_{arf}\cdot XTR_{iart-(f-1)}$$
$$+\sum_{i\in I_{fob}}\sum_{t\in T_u^\ell} shuttime_{ia}\cdot XS_{iat}+\sum_{t\in T_u^\ell}\sum_{r\in RB_b}\sum_{f\in FT} flttime_{arf}\cdot Y_{art-(f-1)}$$
$$+I(a\in A_{tkr})\cdot\left[\sum_{b\in B_{tkr}}\sum_{b'\in B_{arp}}\sum_{t\in T_u^\ell} tkrtime_{abb'}\cdot TKRA_{abb't}\right.$$
$$+\sum_{b\in B_e}\sum_{t\in T_u^\ell} hrsper\cdot rttrv_{ab}\cdot TKREC_{abt}$$
$$\left.+\sum_{b\in B_{tkr}}\sum_{t\in T_u^\ell} hrsper\cdot rttrv_{ab}\cdot TKRBC_{abt}\right]$$
$$\leq\sum_{t\in T_u^\ell} cumac_{at}\cdot urate_a \qquad \forall a\in A$$

The lack of utilization block specificity allows some modeling freedom, hence we re-define these blocks as the active set: $T_u^\ell \equiv TRL^\ell$. Because missions launched in a period usually consume flight time in subsequent periods, the UTE constraint still overlaps the previous subproblem, and must be relaxed. However, since the majority of the associated columns are indexed by the active set (plus some indexed by the extended set), enforcing UTE$^{\ell}$ provides nearly the same restriction on the feasible region.

ACCONSUME$^{\ell}$: max acft usage to lessen rounding effects

$$\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} msntime_{arf}\cdot XD_{iart-(f-1)}\cdot I\left(t-(f-1)\in TRL^{\ell}\cup TO^{\ell}\right)$$

$$+\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} msntime_{arf}\cdot XT_{iart-(f-1)}\cdot I\left(t-(f-1)\in TRL^{\ell}\cup TO^{\ell}\right)$$

$$\sum_{i\in I}\sum_{r\in RD_{ia,dir}}\sum_{f\in FT} msntime_{arf}\cdot XDR_{iart-(f-1)}\cdot I\left(t-(f-1)\in TRL^{\ell}\cup TO^{\ell}\right)$$

$$+\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}}\sum_{f\in FT} msntime_{arf}\cdot XTR_{iart-(f-1)}\cdot I\left(t-(f-1)\in TRL^{\ell}\cup TO^{\ell}\right)$$

$$+\sum_{i\in I_{fob}}\frac{hrsper}{shutrate_{ai}}\cdot XS_{iat}$$

$$+\sum_{r\in RB}\sum_{f\in FT} msntime_{arf}\cdot Y_{art-(f-1)}\cdot I\left(t-(f-1)\in TRL^{\ell}\cup TO^{\ell}\right)$$

$$+I(a\in A_{tkr})\cdot\left[\sum_{b\in B_{tkr}}\sum_{b'\in B_{arp}}\frac{hrsper}{tkrrate_{abb'}}\cdot TKRA_{abb't}\right.$$

$$+\sum_{b\in B_e} rttrv_{ab}\cdot hrsper\cdot TKREC_{abt}$$

$$\left.+\sum_{b\in B_{tkr}} rttrv_{ab}\cdot hrsper\cdot TKRBC_{abt}\right]$$

$$+\sum_{b\in B_e} hrsper\cdot RON_{abt}+\sum_{b\in B_{sup}} hrsper\cdot[RONT_{abt}+RONR_{abt}]$$

$$\leq hrsper\cdot cumac_{at} \qquad \forall a\in A, t\in TRL^{\ell}\cup TO^{\ell}$$

The cascade modification activates rows indexed by the extended-active set, columns indexed by the active set and duplicates columns indexed by the extended set.

DCAPACITY$^{\ell}$: aircraft capacity for direct delivery

$$\sum_{c \in C_a \cap CC} \frac{DTONS_{iact}}{purecap_{iac}} + \frac{paxfrac_a \cdot DTONS_{i,a,pax,t}}{maxpax_a} \cdot I(a \in A_{pax})$$
$$\leq \sum_{r \in RD_{ia,dir}} rangefac_{iar} \cdot [XD_{iart-trv_{ar}} + XDR_{iart-trv_{ar}}]$$
$$\forall i \in I, a \in A, t \in TRL^{\ell} \cap TW_i$$

The cascade modification activates rows and columns indexed by the active set, and duplicates columns indexed by the extended set.

TCAPACITY$^{\ell}$: aircraft capacity for transshipments

$$\sum_{c \in C_a \cap CC} \frac{TTONS_{iact}}{purecap_{iac}} + \frac{paxfrac_a \cdot TTONS_{i,a,pax,t}}{maxpax_a} \cdot I(a \in A_{pax})$$
$$\leq \sum_{r \in RD_{ia,trn}} rangefac_{iar} \cdot [XT_{iart-trv_{ar}} + XTR_{iart-trv_{ar}}]$$
$$\forall i \in I_{fob}, a \in A, t \in TRL^{\ell} \cap TW_i$$

The cascade modification activates rows and columns indexed by the active set, and duplicates columns indexed by the extended set.

SCAPACITY$^{\ell}$: aircraft capacity for shuttle deliveries

$$\sum_{c \in C_a \cap CC} \frac{STONS_{iact}}{purecap_{iac}} + \frac{paxfrac_a \cdot STONS_{i,a,pax,t}}{maxpax_a} \cdot I(a \in A_{pax})$$
$$\leq srange_{ia} \cdot XS_{iat}$$
$$\forall i \in I_{fob}, a \in A, t \in TRL^{\ell} \cap TW_i$$

The cascade modification activates rows and columns indexed by the active set.

DPAXCAP$^{\ell}$: aircraft capacity for direct delivery of pax

$$DTONS_{i,a,pax,t} \leq \sum_{r \in RD_{ia,dir}} maxpax_a \cdot [XD_{iart-trv_{ar}} + XDR_{iart-trv_{ar}}]$$
$$\forall i \in I, a \in A_{mix}, t \in TRL^{\ell} \cap TW_i$$

The cascade modification activates rows and columns indexed by the active set, and duplicates columns indexed by the extended set.

TPAXCAP$^{\ell}$: aircraft capacity for transshipment of pax

$$TTONS_{i,a,pax,t} \leq \sum_{r \in RD_{ia,trn}} maxpax_a \cdot [XT_{iart-trv_{ar}} + XTR_{iart-trv_{ar}}]$$
$$\forall i \in I_{fob}, a \in A_{mix}, t \in TRL^{\ell} \cap TW_i$$

The cascade modification activates rows and columns indexed by the active set, and duplicates columns indexed by the extended set.

SPAXCAP$^{\ell}$: aircraft capacity for delivery of pax by shuttles

$$STONS_{i,a,pax,t} \leq maxpax_a \cdot XS_{iat} \qquad \forall i \in I_{fob}, a \in A_{mix}, t \in TRL^{\ell} \cap TW_i$$

The cascade modification activates rows and columns indexed by the active set.

MEETDEM$^{\ell}$: meet demand for each line id

$$\sum_{a \in A_c} \sum_{t \in TW_i} DTONS_{iact} + NOGO_{ic}$$
$$+I(i \in I_{fob}) \cdot \left[\sum_{a \in A_c} \sum_{t \in TW_i} STONS_{iact} + \sum_{t \in T} GTONS_{ict} \right] = dem_{ic}$$
$$\forall c \in C, \ \forall i : TW_i \subseteq TRL^{\ell}$$

The cascade modification activates this row only if the line id's delivery window is a subset of the active set. Thus, many of these rows are Lagrange-relaxed, which motivates the following supplemental constraint.

MEETDEM1$^{\ell}$: do not exceed demand during each subproblem

$$\sum_{a\in A_c}\sum_{t\in TW_i} DTONS_{iact} + I(i \in I_{fob})\cdot$$
$$\left[\sum_{a\in A_c}\sum_{t\in TW_i} STONS_{iact} + \sum_{t\in T} GTONS_{ict}\right] \leq dem_{ic}$$
$$\forall c \in C,\ \forall i : \left(TW_i \cap \left[TO^{\ell} \cup TRL^{\ell}\right] \neq \emptyset, TW_i \nsubseteq TRL^{\ell}\right)$$

MEET DEMand 1: Tons delivered can never exceed demand for any subproblem. This bounds the $DTONS$, $STONS$ and $GTONS$ variables when the **MEETDEM** constraint is Lagrange-relaxed. The cascade modification activates this constraint whenever the line id's delivery window includes elements of the active set, unless the **MEETDEM** constraint is active.

TRANSTONS$^{\ell}$: flow balance for transshipped stons

$$\sum_{a\in A_c} TTONS_{iact} = \sum_{a\in A_c} STONS_{iact} + GTONS_{ict}\ \ \forall i \in I_{fob},\ c \in C,\ t \in TW_i \cap TRL^{\ell}$$

Because of the time-index shift in the $GTONS_{ict}$ re-definition, the only cascade modification activates rows and columns indexed by $t \in TW_i \cap TRL^{\ell}$.

INITCREWS$^{\ell=1}$: initialize crew placement

$$\sum_{b\in B_{crw}} SCREWS_{abt} + crewrat_a \cdot \sum_{b\in B_{tkr}} TKRB_{abt}$$
$$= crewrat_a \cdot newac_{at} \qquad \forall a, t = 1$$

The cascade modification activates these rows and columns only in the first sub-problem.

SCREWBAL$^{\ell}$: strategic crew balance of flow

$$
\begin{aligned}
&SCREWS_{abt+1} = SCREWS_{abt} \\
&+\sum_{i\in I}\sum_{r\in RD_{ia,dir}\cap\bar{R}_{b,ori}} [XD_{iart-ctrv_{abr}} + XDR_{iart-ctrv_{abr}}] \\
&+\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}\cap\bar{R}_{b,ori}} [XT_{iart-ctrv_{abr}} + XTR_{iart-ctrv_{abr}}] \\
&+\sum_{r\in RB\cap\bar{R}_{b,ori}} Y_{art-ctrv_{abr}} \\
&-\sum_{i\in I}\sum_{r\in RD_{ia,dir}\cap\bar{R}_{b,dst}} [XD_{iart-etrv_{abr}} + XDR_{iart-etrv_{abr}}] \\
&-\sum_{i\in I_{fob}}\sum_{r\in RD_{ia,trn}\cap\bar{R}_{b,dst}} [XT_{iart-etrv_{abr}} + XTR_{iart-etrv_{abr}}] \\
&-\sum_{r\in RB\cap\bar{R}_{b,dst}} Y_{art-etrv_{abr}} \\
&+I(b\in B_e)\cdot crewrat_a\cdot[TKRCE_{abt-cttrv_{ab}} - TKREC_{abt}] \\
&+I(b\in B_{sup})\cdot \\
&crewrat_a\cdot[THCHOP_{abt-1} - THCHOP_{abt} + I(t=1)\cdot IRONT_{ab}] \\
&+I(b\in BS_{rec})\cdot \\
&crewrat_a\cdot[THCHOPR_{abt-1} - THCHOPR_{abt} + I(t=1)\cdot IRONR_{ab}] \\
&+I(b\in B_e, t\neq 1, newac_{at}>0)\cdot crewrat_a\cdot ALLOC_{abt} \\
&+\sum_{b'\in B_{crw}} DHCREW_{ab'bt-dhtrv_{b'b}} - \sum_{b'\in B_{crw}} DHCREW_{abb't} \\
&\qquad\qquad \forall a,b\in B_{crw},\ \forall t: t\in T, t+1\in TRL^{\ell}
\end{aligned}
$$

As with a proximal cascade, this constraint is defined and indexed on $t+1$. The cascade modification activates rows and columns indexed by the active set, and duplicates columns indexed by the extended set. Additionally, $IRONT_{ab}$ and $IRONR_{ab}$ are active for $t=1$.

MOG$^{\ell}$: airfield capacity

$$\sum_{i\in I}\sum_{a\in A}\sum_{r\in RD_b\cap RD_{ia,dir}} gtime_{abr}\cdot acpkg_{ab}\cdot[XD_{iart-etrv_{abr}}+XDR_{iart-etrv_{abr}}]$$
$$\cdot I(t-etrv_{abr}\in TRL^{\ell}\cup TO^{\ell})$$
$$+\sum_{i\in I_{b,dst}}\sum_{a\in A}\sum_{r\in RD_{ia,dir}} gtime_{abr}\cdot acpkg_{ab}\cdot XD_{iart-trv_{ar}}\cdot I(t-trv_{ar}\in TRL^{\ell}\cup TO^{\ell})$$
$$+\sum_{i\in I_{b,dst}}\sum_{a\in A}\sum_{r\in RD_{ia,dir}} qtime_{abr}\cdot acpkg_{ab}\cdot XDR_{iart-trv_{ar}}\cdot I(t-trv_{ar}\in TRL^{\ell}\cup TO^{\ell})$$
$$+\sum_{i\in I}\sum_{a\in A}\sum_{r\in RD_b\cap RD_{ia,trn}} gtime_{abr}\cdot acpkg_{ab}\cdot[XT_{iart-etrv_{abr}}+XTR_{iart-etrv_{abr}}]$$
$$\cdot I(t-etrv_{abr}\in TRL^{\ell}\cup TO^{\ell})$$
$$+\sum_{i\in I_{b,trn}}\sum_{a\in A}\sum_{r\in RD_{ia,trn}} gtime_{abr}\cdot acpkg_{ab}\cdot XT_{iart-trv_{ar}}\cdot I(t-trv_{ar}\in TRL^{\ell}\cup TO^{\ell})$$
$$+\sum_{i\in I_{b,trn}}\sum_{a\in A}\sum_{r\in RD_{ia,trn}} qtime_{abr}\cdot acpkg_{ab}\cdot XTR_{iart-trv_{ar}}\cdot I(t-trv_{ar}\in TRL^{\ell}\cup TO^{\ell})$$
$$+\sum_{i\in(I_{b,dst}\cap I_{fob})\cup I_{b,trn}}\sum_{a\in A} sgtime_{ab}\cdot acpkg_{ab}\cdot XS_{iat}$$
$$+\sum_{b'\in B_{sup}\cap BS_{b,dwn}}\sum_{a\in A} hrsper\cdot acpkg_{ab}\cdot[THCHOP_{ab't}+THCHOPR_{ab't}]$$
$$+\sum_{a\in A}\sum_{r\in RB_{ab}} gtime_{abr}\cdot acpkg_{ab}\cdot Y_{art-etrv_{abr}}\cdot I(t-etrv_{abr}\in TRL^{\ell}\cup TO^{\ell})$$
$$+I(b\in B_{tkr})\cdot\left[\sum_{a\in A_{tkr}} hrsper\cdot acpkg_{ab}\cdot TKRB_{abt}\right]$$
$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,dir}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XD_{iart-etrv_{abr}}$$
$$\cdot I(t-etrv_{abr}\in TRL^{\ell}\cup TO^{\ell})$$
$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,dir}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XDR_{iart-etrv_{abr}}$$
$$\cdot I(t-etrv_{abr}\in TRL^{\ell}\cup TO^{\ell})$$
$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,trn}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XT_{iart-etrv_{abr}}$$
$$\cdot I(t-etrv_{abr}\in TRL^{\ell}\cup TO^{\ell})$$
$$+\sum_{i\in I}\sum_{a\in A_{rfl}}\sum_{r\in RD_{b,div}\cap RD_{ia,trn}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot XTR_{iart-etrv_{abr}}$$
$$\cdot I(t-etrv_{abr}\in TRL^{\ell}\cup TO^{\ell})$$
$$+\sum_{a\in A_{rfl}}\sum_{r\in RB_{b,div}} dpct_a\cdot gtime_{abr}\cdot acpkg_{ab}\cdot Y_{art-etrv_{abr}}\cdot I(t-etrv_{abr}\in TRL^{\ell}\cup TO^{\ell})$$
$$\leq mog_b\cdot mogeff_b \qquad \forall b\in B\setminus B_{sup}\setminus B_{arp}\setminus B_{way}, t\in TRL^{\ell}\cup TO^{\ell}$$

The cascade modification activates rows indexed by the extended-active set, columns indexed by the active set and duplicates columns indexed by the extended set.

Non-negativity of all variables.

E. NRMO CASCADE RESULTS

Because of its structure and complexity, NRMO is an excellent model to test proximal and Lagrangian cascades. A moderately sized scenario consists of hundreds of line id's; large scenarios can easily overwhelm current computing capabilities. Additionally, the model should produce results that are intentionally myopic, since that is a characteristic of the underlying airlift system.

Three NRMO problem instances are used to test cascade performance. The first problem is the primary test scenario used at NPS to verify and validate air mobility linear programs. We took the remaining two scenarios from an ongoing study by RAND [Stucker and Melody, 1996].

The performance tests measure the effect of three parameters on the proximal-Lagrangian gap. Typically, larger values of the proximal cascade width, *caswid,* proximal cascade overlap, v, and Lagrangian cascade width, *lwid* should all reduce the gap. The tests also examine the effect of these parameters on solution time when both simplex and barrier methods solve the cascades.

Each of the three problem instances is generated by GAMS [Brooke, *et al.,* 1992], and written into MPS format. Additionally, the GAMS output provides a file that maps each row and column to its associated time index. The cascade logic executes in *C* using the *CPLEX* callable library version 3.0 [*CPLEX*, 1994]. A utility translates the solution reported by *CPLEX* to a GAMS compatible format. Unless otherwise noted, the computer used is an IBM RS6000/590 with 512MB of RAM. All times are given in CPU seconds.

1. Notional Southwest Asia Scenario

The notional Southwest Asia scenario was originally designed to test THRUPUT II [Lim, 1994], one of NRMO's predecessors. It includes 21 line id's, 7 aircraft types, 35

routes and 30 time periods. The associated linear program has 4,100 rows, 7,400 columns, 39,000 non-zeros, and a maximum staircase overlap of two periods. In this scenario, a contingency in Southwest Asia (SWA) requires deployment of several Army and Marine Corps brigades from CONUS, 15 Air Force fighter wings from CONUS and Europe, and an Army mechanized division from Europe. The requirement intentionally exceeds delivery capacity in order to strain the system and identify airlift bottlenecks.

Cascade Width	Cascade Overlap	Upper Bound	Lower Bound	%Gap	Proximal Time (sec)	Lagrange Time (sec)	Total Time (sec)
Monolith		**294.1**	**n/a**	**n/a**	**n/a**	**n/a**	**61**
20	5	296.6	286.9	3.4	47	19	66
20	10	294.6	290.0	1.6	57	20	77
20	15	294.1	292.5	0.6	94	18	112
18	5	303.6	262.0	15.9	46	18	64
18	10	296.7	287.1	3.3	67	21	88
18	15	294.1	291.6	0.9	124	18	142
15	5	303.3	275.9	9.9	41	19	60
15	10	295.4	286.3	3.2	73	19	92
15	12	294.7	285.2	3.3	107	19	126
10	5	305.3	273.7	11.6	41	20	61
10	7	300.0	266.4	12.6	58	20	78

Table 1. Relative gaps and solution times for the Southwest Asia scenario vary with cascade parameter selection. The first two columns show proximal cascade widths and overlaps; Lagrangian cascade widths are all 15. The remaining columns show the performance (computing times are in seconds on an IBM RS6000/590 with 512MB RAM). For example, a proximal cascade with width 18 and overlap 10 gives an upper bound solution value of 296.7; the corresponding Lagrangian lower bound is 287.1, resulting in a gap of 3.3%. The proximal and Lagrange solve times are 67 and 21 seconds, respectively, for a total of 88 seconds. The first row of the table gives the monolith's solution value and time, which provides a baseline for the other runs. Each test uses *CPLEX 3.0* [*CPLEX, 1994*] with primal simplex method and steepest edge pricing.

Table 1, and Figures 6 and 7 illustrate that solution quality improves with increased cascade overlap and width. Figure 6 shows a strictly decreasing gap with increasing cascade overlap for cascade widths of 18 and 20. These decreasing gaps come at a computational cost, however, as indicated by the proximal cascade solution times. Figure 7 also shows generally decreasing gaps with increased cascade width, albeit less convincingly.

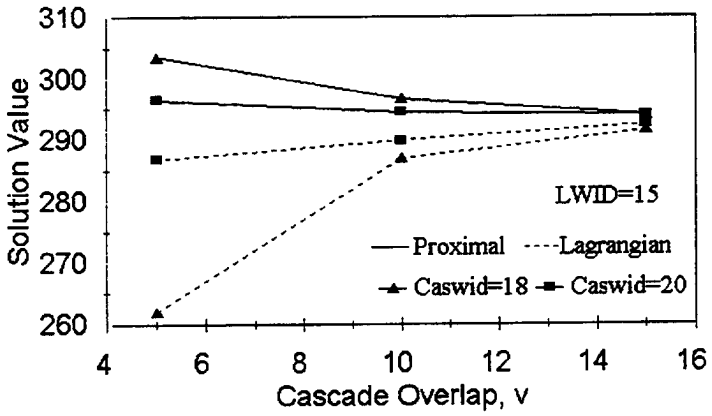

Figure 6. Solution gaps for the Southwest Asia scenario decrease significantly with increased proximal cascade overlap. The triangles show the proximal (solid line) and Lagrangian (dotted line) cascade solution values for an 18 period proximal cascade width; the squares show the solution values for a 20 period width. All Lagrangian cascade widths are 15. The absolute gap, measured by the vertical distance between proximal and Lagrangian solution values, is much smaller with a 10 period overlap than a 5 period overlap, and smaller still for a 15 period overlap.

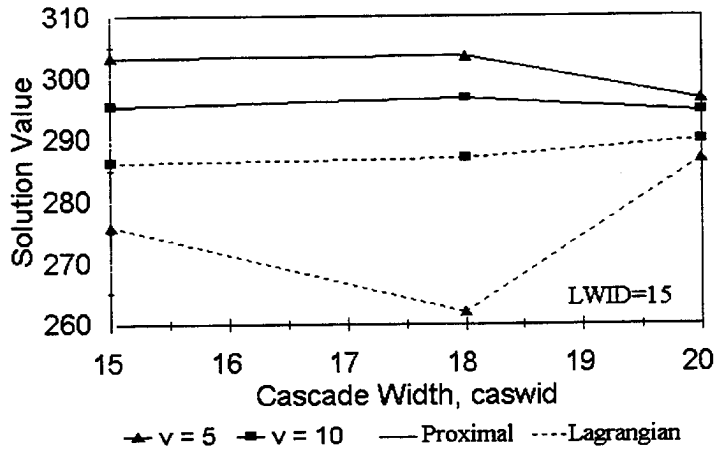

Figure 7. Solution gaps for the SouthwestAsia scenario generally decrease as proximal cascade width increases. Proximal cascade width has a smaller effect on the absolute gap than the proximal cascade overlap.

Cascade Width	Cascade Overlap	Upper Bound	Lower Bound	%Gap	Proximal Time (sec)	Lagrange Time (sec)	Total Time (sec)
Monolith		**294.1**	**n/a**	**n/a**	**n/a**	**n/a**	**61**
20	5	296.6	286.4	3.6	47	11	58
20	10	294.6	288.7	2.1	57	10	67
20	15	294.1	288.1	2.1	94	11	105
18	5	303.6	273.2	11.2	46	11	57
18	10	296.7	279.6	6.1	67	11	78
18	15	294.1	281.4	4.5	124	10	134
15	5	303.3	267.8	13.3	41	11	52
15	10	295.4	281.3	5.1	73	10	83
15	12	294.7	280.2	5.1	107	11	118
10	5	305.3	265.8	14.8	41	11	52
10	7	300.0	263.6	13.8	58	13	71

Table 2. This table depicts Southwest Asia scenario results with the Lagrangian cascade width equal to 10. These results are similar to the $lwid = 15$ test (see the previous table), although the gaps are slightly larger. This is due to the greater number of Lagrange-relaxed rows. Note that the Lagrangian cascade solves faster with three subproblems (this table) than two subproblems (previous table).

Table 2 depicts the SWA scenario results using the same proximal cascade parameters as Table 1, but with a Lagrangian cascade width (*lwid*) of 10. As expected, the lower bounds are weaker for $lwid = 10$ than for $lwid = 15$, because the monolith is split in two places for this relaxation. However, the Lagrange solution times when $lwid = 10$ are about half as long as their $lwid = 15$ counterparts. This is despite the fact that the $lwid = 10$ cascade requires one more subproblem than a $lwid = 15$ cascade.

In this problem, temporal myopia has only a minor effect on solution quality even when the scheduling horizon is reduced by as much as two-thirds. When the proximal cascade width is only 10 periods, the solution values are still within 4% of the monolith solution. Longer solution horizons produce even closer results.

2. European Infrastructure Scenario I

Concurrent with this research, a RAND Corporation study for the Office of the Secretary of Defense (OSD) is examining European air bases transited by USAF airlifters. The

purpose of this study is to determine which bases have insufficient infrastructure to adequately support a Major Regional Contingency (MRC) in Southwest Asia [Stucker and Melody, 1996]. The problem consists of 220 line id's, six aircraft types, 22 routes, and 30 time periods. Approximately 75% of the scenario's movement requirements originate in CONUS. The corresponding linear program has 27,000 rows, 126,500 columns, 921,500 non-zeros, and a maximum staircase overlap of two periods.

Cascade Width	Cascade Overlap	Upper Bound	Lower Bound	%Gap	Proximal Time (sec)	Lagrange Time (sec)	Total Time (sec)
Monolith		**106.1**	**n/a**	**n/a**	**n/a**	**n/a**	**980**
20	5	108.7	93.8	15.8	1010	590	1600
20	10	106.9	101.8	5.0	1260	704	1964
20	15	106.9	102.9	3.9	1907	663	2570
18	5	107.4	91.8	17.0	933	630	1563
18	10	107.6	98.3	9.5	1352	605	1957
18	15	107.1	100.4	6.7	2652	715	3367
15	5	109.2	84.5	29.3	959	659	1618
15	10	107.6	96.0	12.1	1527	650	2177
15	12	107.5	99.9	7.6	2307	601	2908
10	5	113.3	75.8	49.4	1061	639	1700
10	7	110.9	83.7	32.4	1483	770	2253

Table 3. Computational results for European Infrastructure I also show that relative gaps and solution times vary with cascade parameter selection. The solve times are much longer than Southewst Asia scenario solve times due to problem size. All runs use the *CPLEX 3.0* Barrier algorithm [*CPLEX*, 1994]. Lagrangian cascade subproblems have 15 periods each. The first row is the monolith baseline; subsequent rows show performance using various proximal cascade parameters. All times are in seconds.

The results of this scenario (see Table 3, and Figures 8 and 9) are generally consistent with those of the first test. Figure 8 shows a pronounced reduction in gap as cascade overlap increases, while Figure 9 shows a more moderate reduction with increased cascade length. Upper bounds are of better quality than lower bounds, due to the sensitivity of the lower bound to small errors in the Lagrangian penalties. Thus, the proximal cascade results show

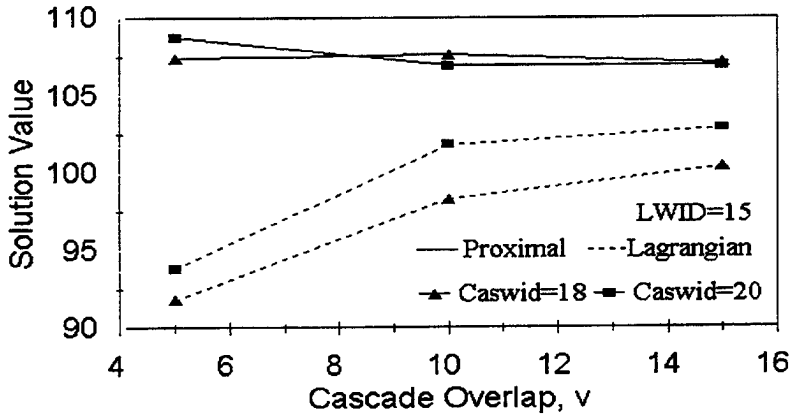

Figure 8. This Figure depicts cascade solution values for the European I scenario when proximal cascade overlap is varied. Proximal cascade overlap has as large an effect on this scenario as it did on the notional Southwest Asia scenario. As before, increasing the overlap reduces the gap.

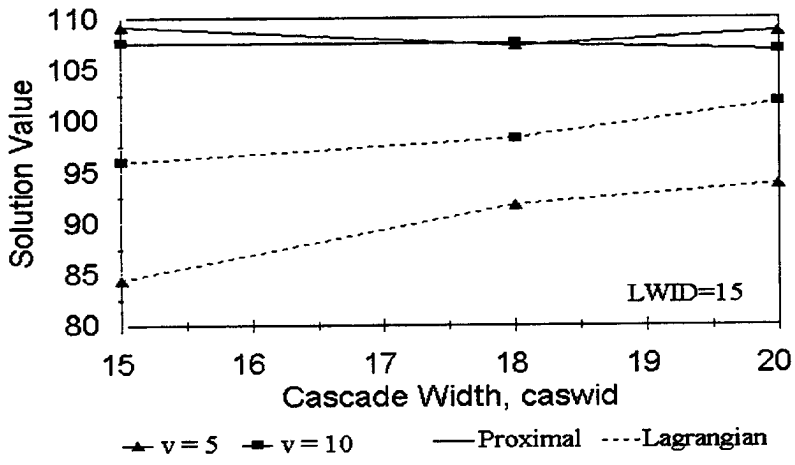

Figure 9. Solution gaps for the European I scenario are reduced with increasing proximal cascade width. These reductions, although smaller than those seen in the Southwest Asia scenario, are still quite evident.

that the effects of myopia are small, since most of the upper bound solution values are within a few percent of the monolith value.

3. European Infrastructure Scenario II

This scenario is a continuation of the RAND study for OSD. However, it includes different assumptions regarding international overflight permissions, and includes a larger Civil Reserve Air Fleet (CRAF) component. As before, the problem consists of 220 line id's and 30 time periods, but there are now eight aircraft types and 24 routes. Additionally, the routes are generally more circuitous than those of European Infrastructure I. The cor-

Cascade Width	Cascade Overlap	Upper Bound	Lower Bound	% Gap	Proximal Time (sec)	Lagrange Time (sec)	Total Time (sec)
Monolith		**247.3**	**n/a**	**n/a**	**n/a**	**n/a**	**860**
20	5	266.2	155.7	71.0	726	391	1117
20	10	251.1	204.9	22.5	926	391	1317
20	15	248.4	206.7	20.8	1399	353	1752
18	5	263.4	156.2	68.6	878	400	1278
18	10	252.0	173.8	45.0	1297	397	1694
18	15	249.5	176.3	41.5	2087	349	2436
15	5	282.5	148.4	90.4	747	352	1099
15	10	262.8	160.1	64.2	1247	373	1620
15	12	255.4	176.8	74.0	1909	378	2287

Table 4. The European Infrastructure II scenario produces considerably larger gaps than the first infrastructure scenario. Much of the difference results from a weaker Lagrangian bound. As before, the proximal and Lagrangian cascades use the *CPLEX 3.0* Barrier algorithm [*CPLEX*, 1994]. All Lagrangian cascades have 15 periods each. Times are in seconds.

responding linear program has 29,400 rows, 115,700 columns, 901,600 non-zeros, and a maximum staircase overlap of two periods.

Table 4 shows the results for a variety of proximal cascade widths and overlaps. These results demonstrate that this problem instance is more affected by myopia than the first two. Consequently, the smallest gap computed is 20.8%, although the largest part of that gap results from a loose lower bound.

4. Solve Time Performance

a. Cascade Versus Monolith

The three scenarios do not exhibit pronounced time savings when using cascades. However, the test platform is a computer with sufficient memory for monolith solution without paging. In order to verify that cascades save time when memory is limited, we reduce the problem size of the two European scenarios by limiting line id delivery windows. This reduction allows solution by a Dell Pentium Pro 200 MHz desktop computer with 64 MB RAM.

Table 5 shows that cascades save up to 80% of the time required for monolith solution. The savings come at a moderate cost in solution quality, since limited memory requires that cascade subproblems have small widths. This consequence is minor in models such as NRMO, where myopia should be enforced regardless of available memory.

Cascade Width	Cascade Overlap	Upper Bound	Lower Bound	%Gap	Proximal Seconds	Lagrange Seconds	Total Seconds	% Time Savings
Reduced European Infrastructure I (14,442 rows, 64,252 columns, 462,645 non-zeros):								
Monolith		**106.9**	**n/a**	**n/a**	**n/a**	**n/a**	**4410**	**n/a**
10	5	116.7	90.2	29.4	572	310	882	80.0
10	7	115.3	92.0	25.2	844	310	1154	73.8
15	5	109.6	96.1	14.1	4080	310	4390	0.5
Reduced European Infrastructure II (16,874 rows, 63,336 columns, 453,663 non-zeros):								
Monolith		**239.7**	**n/a**	**n/a**	**n/a**	**n/a**	**4169**	**n/a**
10	5	245.7	178.4	37.7	532	476	1008	75.8
10	7	243.0	203.5	19.4	760	480	1240	70.3
15	5	242.9	218.1	11.4	2160	480	2640	36.7

Table 5. Cascades offer a significant time savings when the monolith cannot be solved with installed memory. The computer used for these results is a Pentium Pro 200 MHz desktop with 64 MB RAM (previous results use an IBM RS6000/590 with 512 MB RAM). The first row of each scenario shows the monolith solution value and time using the *CPLEX* interactive barrier solver [*CPLEX*, 1994]. The next two rows in each scenario indicate cascades offer a dramatic time savings when moderate cascade widths are used. The final row of each scenario shows that much or all of this savings is lost when cascades also require paging.

b. Barrier Versus Simplex

The barrier algorithm solves cascades of the test scenarios faster than the simplex algorithm, even when large cascade overlaps permit the exploitation of advanced simplex bases. Figure 10 depicts solution speeds for proximal cascades with different numbers of subproblems using the Notional SWA scenario on the IBM RS6000/590. The first

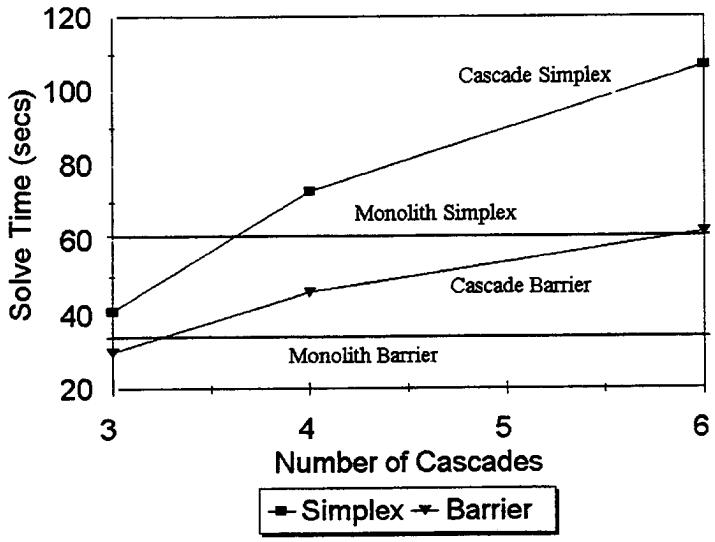

Figure 10. Large problems solve much faster using the barrier algorithm (IBM RS6000/590). The vertical axis shows the notional Southwest Asia cascade solution time in seconds; horizontal bands represent the monolith solution time. The horizontal axis shows the number of subproblems, which is a function of cascade width, *caswid*, and cascade overlap, v. The plots represent the simplex and barrier times with *caswid* fixed at 15, and $v = 5, 10$, and 12. These parameter settings specify the number of subproblems to be 3, 4, and 6, respectively.

impression gleaned from the figure is the disparity between simplex and barrier solve times. This is not surprising, given that *CPLEX* recommends the barrier for problems with more than 1,000 rows and columns [Klotz, 1996]. However, the relative trend of the solve times is surprising. Compared with the simplex cascade, the barrier cascade appears to perform better as the number of subproblems increases, as shown by the divergent trend of the two plots. This is inconsistent with the notion that simplex cascade subproblems are fully exploiting advanced bases. If that were true, simplex performance would improve (relative

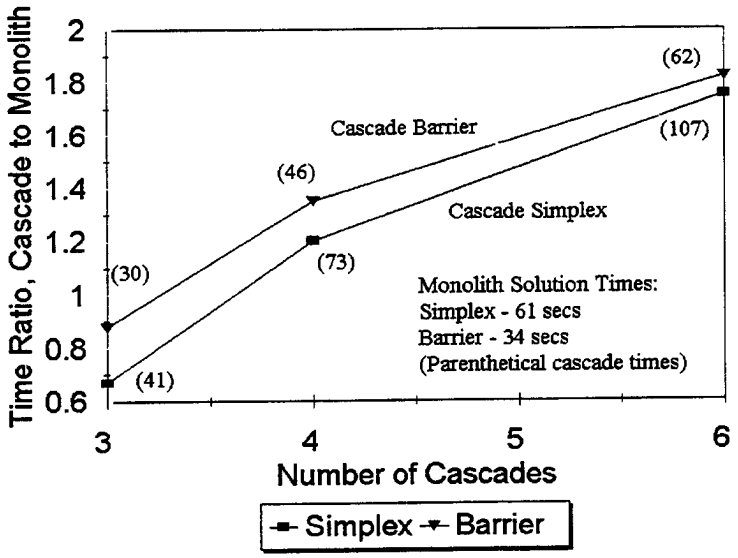

Figure 11. This plot includes the same data as in the previous figure (Notional Southwest Asia, *caswid*=15), but the vertical axis depicts the solve times as a ratio of the cascade solution to the monolith solution. Because the simplex ratios are lower than the barrier ratios, this figure shows that simplex cascades perform better than barrier cascades when compared to their respective monolith solution times.

to the barrier method) as the overlap increases, since more of the basis is preserved from subproblem to subproblem.

Although exploiting advanced bases during the proximal cascade does not appear to be effective, there is one encouraging simplex performance measure. Relative to their respective monolith solve times, a simplex cascade appears to solve faster than a barrier cascade. Figure 11 illustrates this by depicting the vertical axis as the ratio of cascade to monolith solve times for both simplex and barrier. This result is consistent with the idea that simplex solve times increase faster with problem size than do interior point method solve times, since solving the large monolith is relatively more time consuming for the simplex method. Thus, cascades appear more attractive when a barrier algorithm is unavailable.

F. NRMO SUMMARY

The NPS/RAND Mobility Optimization is the most detailed military mobility optimization model ever built. It incorporates all the features of prior models from NPS and

RAND, as well as some additional features. As such, it is a huge model, and solving the monolith may not always be possible. Moreover, NRMO models an imperfect scheduling process, and therefore should incorporate myopia. For these reasons, it provides the motivation and initial test platform for the combined proximal and Lagrangian cascades.

NRMO exhibits the basic staircase structure required by proximal and Lagrangian cascades, but is complicated by numerous additional constraint types. The NRMO formulations in this chapter illustrate how cascades can accommodate a wide variety of constraint types, although minor alteration is sometimes required.

Upper bounds from the proximal cascade are typically within a few percent of monolith optimal. Lower bounds from the Lagrangian cascade have generally less quality, but are often still within a few percent of monolith optimal. Cascade solution times are less than the monolith solution times when small cascade overlaps are used, or when installed memory is limited.

With the cascades now demonstrated on a large and complex model, some generalization is warranted. That generalization is the subject of the next chapter.

V. USING CASCADES WITH GENERAL LPs

We now examine the application of cascades to general linear programs. Specifically, we address what conditions make cascading desirable, and how to improve cascade solution quality. Of foremost concern is assessing cascade *suitability, i.e.,* whether or not a cascade solution is feasible and likely to approximate the monolith solution. A staircase structure with minimal row width (width is the range of the non-null cascade set indices appearing in a row) is perhaps the best indicator of suitability, because all columns associated with each row are proximally related. We propose a simple heuristic to gauge a model's staircase structure by examining a temporal, spatial, or other ordering of rows and columns. Next, we consider some motivations for cascading: 1) inability to solve the monolith due to its large size, 2) desire to intentionally induce solution "myopia," and 3) isolation of subproblems that may solve the monolith faster. Cascades are *appropriate* if a suitable model exhibits, or can be reformulated to exhibit, any of these. Finally, we offer several methods to incorporate dual information into a proximal cascade, thereby reducing the gap between proximal and Lagrangian cascade solution values.

A. WHEN WILL CASCADES WORK?

An arbitrary model monolith may or may not be susceptible to cascade solution. This section offers a method to select a cascade index set that may facilitate cascade feasibility and achieve a good solution. We also examine several model constructs that may reduce cascade suitability, and suggest monolith reformulations that are more amenable to cascades.

1. Gauges for Cascade Suitability

In order to determine suitability, we develop several gauges that can either be used to evaluate a candidate cascade index scheme, or to assess a reformulation to enhance cascade suitability. Assessing suitability *a priori* requires a cascade index set that prescribes an

ordering of rows and columns. The key is to choose an index set where the maximum and average row widths are small; an arbitrary ordering is not likely to exhibit this property. Although time is usually the most intuitive choice for the cascade index, location or priority may also be good choices. Ordering by one of these index sets is likely to reveal the staircase structure in a monolith, if such a structure exists.

We offer the following gauges that suggest suitability of a linear program: 1) the *cascade factor*, $casfactor_T$, which is the average row width normalized for the non-null cardinality of each candidate index set, 2) the *maximum width factor*, w_T, which is the maximum normalized row width, and 3) the *always active rows*, $allact_T$, which is the number of rows that have no correspondence with non-null cascade indices. The following definitions and notation are useful:

- Model rows and columns have one or more candidate cascade index sets. Each set $T \equiv \{0\} \cup \{1, 2, ..., |T|\}$ is composed of a null element and a non-null ordinal subset. An example index is time, where 0 is the null index of a row or column without a time-period index.
- Let model rows be labeled by $i = \{1, 2, ..., |I|\}$, model columns be labeled by $j = \{1, 2, ..., |J|\}$.
- Let a_{ij} be the coefficient in row i and column j.
- Let t_j be the index from set T in column j.
- Define $maxt_{iT}$ as the maximum column index t_j associated with row i.
- Define $mint_{iT}$ as the minimum non-null column index t_j associated with row i. $mint_{iT} = 0$ if all associated columns are null indexed.
- Let $wtotal_T = \sum_i (maxt_{iT} - mint_{iT})$, the sum of row widths.
- Let $averagew_T = wtotal_T \ / \ |I|$, the mean width of all rows.
- Define $straddle_{tT}$ as the number of rows containing non-null elements of T such that $maxt_{iT} > t$ and $mint_{iT} \leq t$.

With these definitions, we can compute the suitability gauges:

$$casfactor_T \quad \begin{cases} averagew_T \ / \ |T| \text{, if } \min_t [straddle_{tT}] > 0 \\ 0, \text{ if } \min_t [straddle_{tT}] = 0 \end{cases}$$

$$w_T \quad (\max_i [maxt_{iT} - mint_{iT}]) \ / \ |T|$$

$$allact_T \quad \text{the number of labels } i \text{ such that } maxt_{iT} = mint_{iT} = 0$$

Of these three gauges, cascade factor (*casfactor*$_T$) is the most comprehensive indicator of cascade suitability, since it considers every row's non-null width. *casfactor*$_T$ is zero if the problem is entirely separable into two or more subsets of T, which implies that proximal cascade subproblems may be solved without loss of monolith optimality. *casfactor*$_T$ does not attempt to distinguish the relative size or number of separable subproblems.

Proximal and Lagrangian cascade solution qualities are conjectured to be better when *casfactor*$_T$ (greater than 0), w_T, and *allact*$_T$ are small. Small row width suggests fewer fixed columns and fewer Lagrange-relaxed rows in the proximal and Lagrangian cascades.

Maximum row width is also an important indicator of cascade suitability. A single row that links all non-null indices may result in an infeasible cascade or a solution of low quality, since satisfying this row may require that all associated columns be simultaneously active. Consequently, smaller values of w_T should indicate better cascade suitability.

Finally, *allact*$_T$ reports the number of rows whose associated columns have null index $t_j = 0$. These rows must be handled by exception when forming cascade subproblems, since they are not accommodated by the ordering prescribed by the candidate index set. As with *casfactor*$_T$ and w_T, smaller values of *allact*$_T$ should indicate better cascade suitability, since fewer exceptions must be dealt with.

Although these gauges may correctly predict cascade suitability, there are some model constructs that cause them to give an incorrect assessment. These gauges can also indicate the presence of model constructs that may be altered to increase cascade suitability. We discuss several of these constructs below.

2. Cumulant Constraints Complicate Suitability

Small formulation differences can have a marked effect on cascade suitability. Consider the *staircase* form of a production-inventory constraint:

$$X_t + I_{t-1} - I_t = d_t \quad \forall t,$$

where X_t, and I_t, represent production and inventory decision levels in order to satisfy a

specified demand, d_t. When defined for a contiguous set of time periods, each constraint overlaps its predecessor by one period.

Production-inventory constraints can alternatively be written in a *cumulant* form [*e.g.*, Johnson and Montgomery, 1974, pp. 197-199]:

$$\sum_{t' \le t} X_{t'} \ge \sum_{t' \le t} d_{t'} \quad \forall t.$$

These two almost equivalent forms yield different values of $casfactor_T$. Each constraint of the staircase form has width two. In contrast, cumulant constraints have a width increasing from 1 to $|T|$. Although a proximal cascade is unaffected, the cumulant form may lower the associated Lagrangian cascade's solution quality because more rows must be Lagrange-relaxed. Reformulating cumulants as staircase constraints for the Lagrangian cascade provides the simplest redress.

Even though $casfactor_T$ gives a warning when cumulant constraints are present, the cumulant form may improve computational efficiency. Consider these two basis matrices:

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

The basis S arises from a set of tight staircase production inventory constraints, while the basis C derives from an equivalent set of cumulant constraints. Note that $S = C^{-1}$. Because of sparseness of the inverse, solver computations could be significantly reduced by cumulant constraints.

3. Rows that are Always-Active

A proximal cascade relies on a cascade index set that has few rows active in every subproblem. Rows whose associated columns all have null cascade index "0" are always-active and are tallied by the $allact_T$ gauge. Additionally, a row associated with *any* column with null index "0" must be active until the null-indexed column is fixed. Inactivating this

row prior to fixing the null-indexed column allows subsequent subproblems to alter the column's level, possibly violating the inactive row.

To illustrate how always-active constraints can reduce cascade suitability, consider a modification of problem S from Chapter II. This modification includes some intermediate time period τ, and variables A and B that are bounded by d in an always-active constraint.

$$Z^S = \min \sum_{t \in T} h_t X_t$$

$$\begin{aligned} s.t. \quad & \sum_{t' \in TS_t} a_{tt'} X_{t'} + A \geq s_t \quad \forall 1 \leq t \leq \tau \\ & \sum_{t' \in TS_t} a_{tt'} X_{t'} + B \geq s_t \quad \forall \tau < t \leq |T| \\ & A + B \leq d \\ & X_t \geq 0 \; \forall t \in T \quad A, B \geq 0. \end{aligned}$$

Define the proximal cascade subproblems as

$$Z^n = \sum_{n' < n} \sum_{t \in TF^{n'}} h_t X_t^{n'} + \min \sum_{t \in TC^n} h_t X_t$$

$$\begin{aligned} s.t. \quad & \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} + A \geq s_t - \sum_{n' < n} \sum_{t' \in TS_t \cap TF^{n'}} a_{tt'} X_{t'}^{n'} \quad \forall t \in TC^n, t \leq \tau \quad (S1.1) \\ & \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} + B \geq s_t - \sum_{n' < n} \sum_{t' \in TS_t \cap TF^{n'}} a_{tt'} X_{t'}^{n'} \quad \forall t \in TC^n, t > \tau \quad (S1.2) \\ & A + B \leq d \quad (S1.AA) \\ & X_t \geq 0 \; \forall t \in TC^n \quad A, B \geq 0. \end{aligned}$$

Constraint *S1.AA* must be active until both A and B are fixed, which occurs in the last subproblem. Furthermore, each row of constraint set *S1.1* must be active until A is fixed. A similar condition holds with B for each row of *S1.2*.

This example illustrates that always-active rows may or may not affect cascade suitability. In some cases, always-active rows may cause solution time to be increased modestly by enlarging each associated subproblem. For example, if the always-active row is redundant, and a presolver does not detect the redundancy, the only negative effect is increased solution time. But in other cases, always-active rows may inflict infeasibility on the later

proximal cascade subproblems because their inactive columns have already been fixed by earlier subproblems. This could occur if, for example, problem S above has only one feasible solution, and that solution includes $A = 0$, and $B = d$. In that case, a cascade that fixes $A > 0$ as a result of early subproblems will culminate with an infeasibility.

Lagrangian cascade solutions will typically have better quality when few rows are "always-active." A proximal cascade row that is always-active must be "always Lagrange-relaxed" in the Lagrangian cascade to preserve the Lagrangian bound for the monolith. If optimal Lagrange penalties are not known, low solution quality may result.

4. Special Conditions in the First and Last Subproblems

Linear programs with time, priority or other candidate ordinal cascade index set may have special boundary conditions, such as specification of inventory before the first or after the last period. Consequently, the first or last subproblem may have unique variable or constraint blocks associated with these boundary conditions. Although starting conditions generally do not affect cascade suitability, ending conditions may result in cascade infeasibility due to myopia. In this case, a model reformulation to include *elastic persistent constraints* [Brown, Dell, and Wood, forthcoming] may redress the infeasibility. We suggest this approach could be applied to a proximal cascade in a production-scheduling LP, for example. These constraints would penalize deviation from target inventory values at the end of each subproblem. The target values would be specified to approach over time the required inventory of the last period, which is strictly enforced.

The gauges developed in this section suggest cascade suitability of a general LP by evaluating three pertinent model characteristics. As we have shown, however, caution must be taken when using them.

B. WHEN ARE CASCADES APPROPRIATE?

Cascades are appropriate if monolith structure is suitable *and* there is sufficient reason to warrant any loss of monolith optimality. The following sections outline three conditions where a suitable model should be cascaded.

1. Cascades used with Large Problems

We use cascades of large problems to reduce solution time. Empirically, solution time for a linear program increases super-linearly with problem size. A cascade reduces solution times (thereby allowing larger problems) by breaking the monolith into smaller subproblems. The cascade implementation of NRMO supports this conjecture. When the cascade overlaps are small, both proximal and Lagrangian cascades are solved using roughly the same time required by the monolith. The NRMO tests also show a dramatic time savings when each subproblem solves using only installed memory and the monolith solution requires disk "paging." In this case, the combined proximal and Lagrangian cascades solve in much less time than the monolith.

2. Cascades to Induce Myopia

Cascades can be used to ensure models do not presume knowledge that is unavailable due to temporal, spatial, or other "remoteness," *i.e.*, lack of proximity. This is the case with NRMO, which models a myopic scheduling process. Conversely, enforcing myopia without a cascade is very tedious for any moderately sized problem, as demonstrated below.

Consider how myopia would be expressed in a single monolithic linear program. Primal feasibility is enforced for all the constraints, yet dual feasibility and complementary slackness must hold for each myopic solution sub-horizon. To illustrate, consider a modification of problem S:

$$
\begin{aligned}
(S) \qquad & Z^S = \min \sum_{t \in T} h_t X_t \\
s.t. \quad & \sum_{t' \in TS_t} a_{tt'} X_{t'} \geq s_t \qquad \forall t \in T \\
& X_t \geq 0 \qquad \forall t \in T.
\end{aligned}
$$

Problem $SCAS^n$ is the corresponding proximal cascade formulation:

$$(SCAS^n) \quad Z^n = \sum_{n'<n} \sum_{t \in TF^{n'}} h_t X_t^{n'} + \min \sum_{t \in TC^n} h_t X_t$$

$$s.t. \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \geq s_t - \sum_{n'<n} \sum_{t' \in TS_t \cap TF^{n'}} a_{tt'} X_{t'}^{n'} \quad \forall t \in TC^n$$
$$X_t \geq 0 \qquad \forall t \in TC^n.$$

Consider a 4-period instance of S with a single period overlap, but with the additional stipulation that the X_1 decision be made prior to knowing the values of h_4 and s_4. A proximal cascade formulation $SCAS$ easily incorporates this situation:

$$\begin{array}{lrrrrcll} (SCAS^1) & \min_X & h_1 X_1 & +h_2 X_2 & +h_3 X_3 & & & \\ & s.t & a_{11} X_1 & & & \geq & s_1 & (W_1) \\ & & a_{21} X_1 & +a_{22} X_2 & & \geq & s_2 & (W_2) \\ & & & a_{32} X_2 & +a_{33} X_3 & \geq & s_3 & (W_3) \\ & & X_1, & X_2, & X_3, & \geq & 0 & \end{array}$$

$$\begin{array}{llrrrrcl} (SCAS^2) \quad h_1 X_1^1 + & \min_X & h_2 X_2 & +h_3 X_3 & +h_4 X_4 & & & \\ & s.t & a_{22} X_2 & & & \geq & s_2 - a_{21} X_1^1 & \\ & & a_{32} X_2 & +a_{33} X_3 & & \geq & s_3 & \\ & & & a_{43} X_3 & +a_{44} X_4 & \geq & s_4 & \\ & & X_2, & X_3, & X_4, & \geq & 0. & \end{array}$$

An equivalent monolithic formulation that formally exhibits "myopia" has to satisfy both the dual feasibility and complementary slackness conditions of $SCAS^1$ in addition to the original constraints. This can be expressed by introducing surplus columns R and slack columns L :

$$
\begin{array}{lccccccccc}
(M) & & & & & & & & & \\
\min_{X,R,W,L} & h_1X_1 & +h_2X_2 & +h_3X_3 & +h_4X_4 & & & & & \\
s.t & a_{11}X_1 & & & & -R_1 & = & s_1 & & \\
 & a_{21}X_1 & +a_{22}X_2 & & & -R_2 & = & s_2 & (M.1) & \\
 & & a_{32}X_2 & +a_{33}X_3 & & -R_3 & = & s_3 & & \\
 & & & +a_{43}X_3 & +a_{44}X_4 & & \geq & s_4 & & \\
 & & & & & & & & & \\
 & & a_{11}W_1 & +a_{21}W_2 & & +L_1 & = & h_1 & & \\
 & & & a_{22}W_2 & +a_{32}W_3 & +L_2 & = & h_2 & (M.2) & \\
 & & & & a_{33}W_3 & +L_3 & = & h_3 & &
\end{array}
$$

$$
\begin{array}{lll}
R_1W_1 = 0 & R_2W_2 = 0 & R_3W_3 = 0 \\
L_1X_1 = 0 & L_2X_2 = 0 & L_3X_3 = 0
\end{array}
\qquad (M.3)
$$

$$
X_1,...,X_4 \geq 0, \quad R_1, R_2, R_3 \geq 0, \quad L_1, L_2, L_3 \geq 0. \qquad (M.4)
$$

Two difficulties arise with this formulation, foremost being tractability. Constraint block $M.3$ (complementary slackness) specifies that at most one of each constraint's elements may exist in the solution at any time. This represents a logical condition where "at most one" element is non-zero, and can be enforced with *binary auxiliary variables* [*e.g.,* Hillier and Lieberman, 1986, p. 394]. Alternately, this condition could be imposed by a *complementary pivoting algorithm* [*e.g.,* Bazaraa, Sherali, and Shetty 1993, pp. 493-500], although this is a heuristic. A model with many time horizons may need to enforce myopia for each horizon, with a concomitant increase in the number of constraints.

The remaining difficulty with the above formulation is that it still doesn't completely enforce myopia. In the presence of multiple optima, the dual feasibility and complementary slackness constraints of M allow selection of the "best" periods' 1, 2, and 3 decisions with respect to period 4. A genuinely myopic formulation selects arbitrarily among the first solution horizon's multiple optima, because it has no foresight that allows tie-breaking.

The literature offers no other method of enforcing myopia in a monolithic formulation. Cascades appear to be a very attractive way of modeling this restriction.

3. Cascades to Isolate Nearly Independent Subproblems

Cascades can be used to reduce solution time when independent, or at least not strongly interdependent, subproblems can be isolated and solved very quickly. Although solving subproblems for this purpose is not the focus of this dissertation, using subproblems to produce a crash basis remain a third reason to cascade. We overview this strategy here for completeness.

A classic example of isolating nearly independent subproblems is the multi-commodity network, with lots of easy network subproblems coupled by joint capacity constraints. Solving each network subproblem, ignoring joint capacitation constraints, and then using these subproblem solutions to give an advanced starting solution for the monolith may solve the monolith much faster than a single cold-start solve attempt [*e.g.,* Staniec 1987]. The *CPLEX* solver [1994, pp. 33-35] offers an option to solve a single imbedded pure network and then use the solution to crash the monolith. Clearly, the advantage of such an indirect approach is enhanced if there are many disjoint subproblems (for example, hundreds of commodity networks) and if only a few joint commodity capacitation constraints are actually binding at optimality.

Brown, Graves, and Ronen [1987] use cascades to solve the LP relaxation of large set partitions. Here the cascade index set is not deduced from any model indexing, but must be determined by heuristic topological sorts of the technological coefficients. Once sorted, the cascade initially restricts each subproblem to active variables with non-zero values from prior subproblem solutions, and then relaxes to all active variables. The authors solve larger and larger nested subproblems until the monolith is solved.

C. IMPROVING CASCADES WITH DUAL PRICES

1. Lagrangian Penalties for Proximal Cascades

This section explores how subsequent proximal cascade subproblems may exploit dual information from previous cascades or subproblems. We then demonstrate two such methods on ten small staircase problems.

Until now, dual information from a proximal cascade provided penalties only for the Lagrangian cascade. However, a similar idea extends to a proximal cascade, which also relaxes constraints. In the case of time, a proximal cascade subproblem relaxes the constraints of "future" periods, *i.e.,* the set of periods that are not active in any subproblem $1, ..., n$. Lagrangian penalties can "charge" active columns for future resource usage, transforming relaxed proximal cascade constraints to Lagrange-relaxed constraints. This is similar to the Lagrangian cascade, except Lagrangian subproblems incorporate penalties from both prior and future constraints. A proximal cascade explicitly enforces all constraints by sending fixed primal columns forward to the next subproblem. This is necessary because even optimal Lagrangian penalties do not guarantee that relaxed constraints are satisfied. The Lagrangian cascade is not explicitly altered, although any improvement in the proximal cascade solution quality will be reflected in the associated duals, and therefore the Lagrangian cascade.

To illustrate, consider a modification of problem $SCAS^n$ for some $\beta_t \geq 0$:

$$Z^n = \sum_{n'<n} \sum_{t \in TF^{n'}} h_t X_t^{n'} + \min \sum_{t \in TC^n} h_t X_t + \sum_{lastp^n < t \leq t+m} \beta_t \left(s_t - \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \right)$$

$$s.t. \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \geq s_t - \sum_{n'<n} \sum_{t' \in TS_t \cap TF^{n'}} a_{tt'} X_{t'}^{n'} \quad \forall t \in TC^n$$

$$X_t \geq 0 \qquad \forall t \in TC^n.$$

The additional objective term does not provide a straightforward Lagrangian relaxation, since it only includes active columns, rather than all columns from the original row. However, the formulation makes clear the intention to reward satisfaction of rows in subproblem $n+1$ by columns in subproblem n. By ignoring the constant term, the desired formulation is

$$(SL^n) \qquad Z^n = \sum_{n'<n} \sum_{t \in TF^{n'}} h_t X_t^{n'} + \min \sum_{t \in TC^n} h_t X_t - \sum_{lastp^n < t \leq t+m} \beta_t \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'}$$

$$s.t. \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \geq s_t - \sum_{n'<n} \sum_{t' \in TS_t \cap TF^{n'}} a_{tt'} X_{t'}^{n'} \quad \forall t \in TC^n$$

$$X_t \geq 0 \qquad \forall t \in TC^n.$$

Assuming β_t exists only for $t \in T$, the proximal cascade solution value remains unmodified:

$$Z^N = \sum_{n \in NC} \sum_{t \in TF^n} h_t X_t.$$

This formulation not only enforces feasibility of active rows, but encourages satisfaction of future subproblem rows through the use of some $\beta_t \geq 0$. Grinold [1983] introduces a similar technique for infinite horizon programming, although his formulation also accounts for the contribution of inactive columns, and retains the constant term as a measure of "salvage."

The Lagrangian penalty formulation requires exogenous specification of the Lagrange multiplier, β_t. How one selects a proper value depends on the underlying motivation for the cascade. If omniscience is acceptable and multiple *series* of cascades (proximal and Lagrangian cascades performed more than once) can be made, the multipliers for a subproblem can be taken from the corresponding constraints of an earlier solution of the same subproblem. Each subproblem receives prices from future periods of the previous series, and re-solves based on those prices. On the other hand, if myopia must be enforced, multipliers are passed forward from "similar" constraints of previous subproblems. Computational results using each of these strategies are discussed next.

a. Iterated Lagrange Multipliers

Iterated Lagrange multipliers derive Lagrangian penalties from the previous cascade series. In tests described in this section, these series are run until no further improvement in solution quality occurs.

Instances of model S with 20 periods and varying staircase overlaps m provide a test case for *Iterated Lagrange multipliers.* Problem S is reformulated into subproblems of the form SL^n. The test considers 10 different sets of overlap and cascade parameters. The penalties (β_t) for Lagrange-relaxed constraints in the proximal cascade come from the previous series; the initial series of each set uses $\beta_t = 0$. Subsequent series' penalties derive from the last subproblem in which the corresponding constraints are active. The Lagrangian cascade selects penalties from the most recent proximal cascade subproblem in

which the corresponding constraints are active. Table 6 describes the results.

Set #	m	caswid	v	lwid	Series 1 % gap	Series 2 % gap	Series 3 % gap	Series 4 % gap
1	1	4	3	5	7.0	2.0	2.0	.2
2	1	5	2	5	10.7	-	-	-
3	1	7	2	7	11.5	1.7	2.0	0
4	2	5	4	5	23.2	-	-	-
5	2	6	2	5	22.4	15.2	-	-
6	2	6	4	5	21.5	15.6	1.0	-
7	2	6	5	5	21.2	13.9	0	
8	2	7	4	7	13.2	8.8	-	-
9	2	10	2	10	13.8	0		
10	2	10	4	10	5.7	1.3	0	

Table 6. Sets of cascade SL use various widths and overlaps to test *iterated Lagrange multipliers*. In this test, each pair of proximal and Lagrangian cascades forms a series. The multipliers for proximal cascade Lagrange-relaxed constraints in each subproblem come from the previous series. The initial series' multipliers are 0. Each subsequent series' multiplier comes from the last subproblem in which the corresponding constraint is active. The Lagrangian cascade selects a multiplier from the most recent proximal cascade subproblem in which the corresponding constraint is active. For instance, Set #6 has staircase overlap 2, proximal cascade width 6, proximal cascade overlap 4, and Lagrangian cascade width 5. The series 1 gap of 21.5% reflects no dual information. The series 2 and 3 gaps of 15.6% and 1.0%, respectively reflect new and more accurate multipliers. A "-" series entry indicates that the gap oscillates back to a previous value, and no further improvement occurs. The results show significant gap improvement in all but sets 2 and 4. The mean gap is reduced from 15.0% to 5.9%.

These results show that information regarding future constraints reduces the average proximal-Lagrangian gap from 15.0% to 5.9%. The gap is reduced in 8 of 10 sets.

Incorporating penalties in the proximal cascade objective improves the Lagrangian cascade quality as well. Averaged over the 10 sets, the Lagrangian cascade solution values account for 53% of the total gap reduction, proximal cascades account for the remaining 47%.

Iterated Lagrange multipliers provide encouraging results, but require multiple series of cascades that "look into the future." Consequently, any improved solution quality requires more computation and omniscience.

b. Forward Pass Multipliers

We can pass forward dual multipliers from prior subproblems in a proximal cascade without violating myopia. Myopia does not preclude incorporating past information to better accommodate the future. To the extent that constraints have homogeneous structure from period to period, Lagrange multipliers from previous subproblems may approximate resource consumption penalties of future periods.

We use cascade SL to demonstrate *forward pass multipliers* since it has a homogeneous structure. In this test, each penalty passed to subproblem $n+1$ is the mean of the optimal multipliers from the staircase constraints of the last m periods in subproblem n. For example, if the active periods of subproblem 1 are 1 through 10, and the staircase overlap is 2, the multipliers passed forward to subproblem 2 are the average of the optimal staircase duals from periods 9 and 10.

Set #	Unaltered % gap	Forward Pass Multipliers % gap
1	7.0	6.8
2	10.7	10.7
3	11.5	11.5
4	23.2	12.9
5	22.4	37.6
6	21.5	21.1
7	21.2	21.2
8	13.2	2.7
9	13.8	7.9
10	5.7	3.8

Table 7. Sets of cascade SL use various widths and overlaps to test *forward pass multipliers*. Using the same widths and overlaps as Table 6, sets 1 through 10 are used to test a single cascade series with dual penalties passed forward from each proximal cascade subproblem to its successor. Each penalty passed to subproblem $n+1$ is the mean of the optimal multipliers from constraints of the last m periods in subproblem n. Myopia is not violated, since only "historical" resource prices are used to predict the future. For instance, the gap from set #4 without passing forward multipliers is 23.2%. The gap improves to 12.9% when multipliers are passed forward. Only set 5 produces a larger gap, and the average gap is reduced from 15.0% to 13.6%.

Table 7 describes the results of this test using the same sets from Table 6. The percentage gaps shown are with and without forward pass multipliers.

Incorporating forward pass multipliers yields the same or smaller gap in 9 out of 10 sets. This result is not surprising. Rather than simply ignoring the restrictions imposed by the future, the model predicts the future based on an average assessment of the past.

As with iterated Lagrange multipliers, over half of the forward pass multipliers gap improvement results from better dual information strengthening the Lagrangian bound. On average, 62% of the average gap reduction comes from the Lagrangian cascade; 38% comes from proximal cascade improvement.

Passing forward average values is imprecise; performance can be improved considerably given any underlying knowledge of which past constraints are similar to future constraints. For example, the relative value of resources A and B often remains similar across time. Averaging the two values of subproblem n's marginal cost for A and B makes little sense when forecasting subproblem $n+1$'s penalties. Cyclical similarity also might occur in a model that, for example, has a cascade index set covering many weeks in daily time increments. In this situation, last Friday's multipliers might provide a better forecast for next Friday than an average of last Thursday's, and Friday's, and Saturday's multipliers. Perhaps an idea as simple as exponential moving average duals would capture the sense of proximity using cascades. Applying this technique is model specific, and presents a topic for future research.

2. Explicitly Improving Lagrangian Cascades

Just as traditional Lagrange multiplier search methods tighten the bound provided by Lagrangian relaxation [*e.g.*, Parker and Rardin, pp. 212-237], a Lagrangian cascade also benefits from improved multipliers. The iterated Lagrange multipliers method provides an opportunity for a multiplier search after each series. However, search methods are computationally expensive, so providing this feedback may lengthen the series considerably. Since the iterated Lagrange multipliers method already updates multipliers from series-to-series and improves the Lagrangian bound as a result, the extra effort may not be warranted. Nonetheless, it remains an area for future research.

D. A CASCADING VARIATION OF BENDERS' DECOMPOSITION

In this section, we show that cascade solution quality improves significantly on test problems using a strategy that adds cuts from previous cascade series. These cuts, although weak, yield problem S solution values within a few percent of monolith optimal after several series. This approach is a heuristic variation of Benders' decomposition [Benders' 1962].

The 2-stage Benders' decomposition of Van Slyke and Wets [1969], and its multi-stage extension by Birge [1985] successively add cuts that support the (convex) Lagrangian dual function. Using a variation of this approach in concert with the iterated Lagrange multiplier technique, we attain a tighter gap between the proximal and Lagrangian cascade solutions on 10 simple staircase problems.

In order to demonstrate how Benders' cuts can be incorporated into a proximal cascade, define $TE^n = \{t : t > lastp^n\}$ as the future periods (the set of periods that are not active in subproblems $1, ..., n$). Problem $BCAS^n$ is the solution to the remaining periods, given the fixed columns of subproblems 1,...,$n-1$. In other words, $BCAS^n$ provides the solution to the remaining monolith, given the cascade solution for $t \in \bigcup_{n'<n} TF^{n'}$:

$$(BCAS^n) \qquad Z^n = \sum_{n'<n} \sum_{t \in TF^{n'}} h_t X_t^{n'} + \min_X \sum_{t \in TC^n} h_t X_t + \min_X \sum_{t \in TE^n} h_t X_t$$

$$s.t. \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \geq s_t - \sum_{n'<n} \sum_{t' \in TS_t \cap TF^{n'}} a_{tt'} X_{t'}^{n'} \quad \forall t \in TC^n \qquad (BCAS.1)$$

$$\sum_{t' \in TS_t \cap TE^n} a_{tt'} X_{t'} \geq s_t - \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \quad \forall t \in TE^n$$

$$X_t \geq 0 \qquad \forall t \in TC^n \cup TE^n.$$

Taking the dual of the rows and columns indexed by $t \in TE^n$ yields:

$$Z^n = \sum_{n'<n} \sum_{t \in TF^{n'}} h_t X_t^{n'} + \min_X \sum_{t \in TC^n} h_t X_t + \max_\beta \sum_{t \in TE^n} \beta_t \left(s_t - \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \right)$$

$$s.t. \quad (BCAS.1),$$

$$\sum_{t': t \in TS_{t'}} \beta_{t'} a_{t't} \leq h_t \quad \forall t \in TE^n \qquad (BCn.1)$$

$$X_t \geq 0 \quad \forall t \in TC^n \qquad \beta_t \geq 0 \quad \forall t \in TE^n.$$

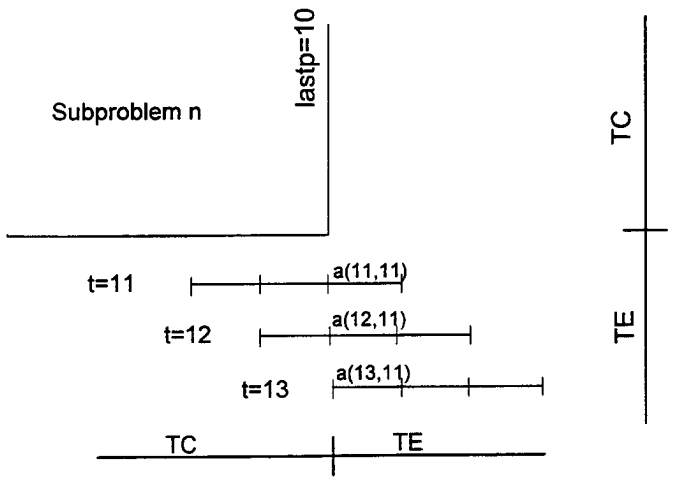

Figure 12. This figure illustrates the terms of a dualized constraint from problem $BCAS^n$, which represents the remaining monolith given the solution to subproblems $1, ..., n-1$. The periods of $BCAS^n$ are partitioned into sets TC^n and TE^n. TC^n has the same width as all previous subproblems; TE^n consists of all the future periods. In the example, the last period of TC^n is 10, the first period of TE^n is 11, and the constraint overlap m is 2. When rows indexed by TE^n are dualized, the left-hand-side terms in the row indexed by period 11 are $\beta_{11}a_{11,11}$, $\beta_{12}a_{12,11}$, and $\beta_{13}a_{13,11}$, or $\sum_{t':11\in TS_{t'}} \beta_{t'}a_{t',11}$.

Refer to Figure 12 for an illustration that describes why the left-hand-side terms in equation BCn.1 are all indexed by $t \in TE^n$.

Proceeding with the Benders' decomposition, the above formulation is equivalent to

$$Z^n = \sum_{n'<n} \sum_{t\in TF^{n'}} h_t X_t^{n'} + \min_X \sum_{t\in TC^n} h_t X_t + \max_{1\le i\le |B|} \sum_{t\in TE^n} \beta_t^{(i)} \left(s_t - \sum_{t'\in TS_t\cap TC^n} a_{tt'} X_{t'} \right)$$

$$s.t. \quad (BCAS.1)$$

$$X_t \ge 0 \qquad \forall t \in TC^n.$$

where $\beta_t^{(i)}$ is a component of vector $\mathbf{b}^{(i)} \in \mathbf{B}$, the set of extreme points defined by the region:

$$\sum_{t':t\in TS_{t'}} \beta_{t'} a_{t't} \le h_t \quad \forall t \in TE^n$$

$$\beta_t \ge 0 \quad \forall t \in TE^n$$

(for simplicity, and to avoid the need for feasibility cuts, we assume that the feasible region in this problem is a bounded polytope [*e.g.*, Parker and Rardin, 1988, pp.237-244]). Thus,

cascade $BCAS^n$ may be rewritten as

$$Z^n = \sum_{n'<n} \sum_{t \in TF^{n'}} h_t X_t^{n'} + \min_{X,\theta} \sum_{t \in TC^n} h_t X_t + \theta^n$$

$$s.t. \quad (BCAS.1)$$

$$\theta^n \geq \sum_{t \in TE^n} \beta_t^{(i)} \left(s_t - \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \right) \qquad i = 1, ..., |\mathbf{B}| \qquad \text{(BCAS.2)}$$

$$X_t \geq 0 \qquad \forall t \in TC^n.$$

Each of these proximal cascade subproblems serves as the master problem for its successor and the subproblem of its predecessor. The Benders' subproblem consists of deriving additional cuts of the form given by $BCAS.2$. A subset of these constraints approximates $BCAS^n$, which is an approximation of the monolith when $n = 1$.

Cut generation for $BCAS^n$ is done by $BCAS^{n+1}$. Unfortunately, instead of the subproblem implied by $BCAS.2$, the available subproblem provided by $BCAS^{n+1}$ is

$$(BCAS^{n+1}) \qquad Z^n = \sum_{n'<n+1} \sum_{t \in TF^{n'}} h_t X_t^{n'} + \min_X \sum_{t \in TC^{n+1}} h_t X_t + \min_X \sum_{t \in TE^{n+1}} h_t X_t$$

$$s.t. \quad \sum_{t' \in TS_t \cap TC^{n+1}} a_{tt'} X_{t'} \geq s_t - \sum_{n'<n+1} \sum_{t' \in TS_t \cap TF^{n'}} a_{tt'} X_{t'}^{n'} \qquad \forall t \in TC^{n+1}$$

$$\sum_{t' \in TS_t \cap TE^{n+1}} a_{tt'} X_{t'} \geq s_t - \sum_{t' \in TS_t \cap TC^{n+1}} a_{tt'} X_{t'} \qquad \forall t \in TE^{n+1}$$

$$X_t \geq 0 \qquad \forall t \in TC^{n+1} \cup TE^{n+1}.$$

Taking the dual of $BCAS^{n+1}$ yields

$$Z^n = \sum_{n'<n+1} \sum_{t \in TF^{n'}} h_t X_t^{n'}$$

$$+ \max_{\beta} \sum_{t \in TC^{n+1}} \beta_t \left(s_t - \sum_{n'<n+1} \sum_{t' \in TS_t \cap TF^{n'}} a_{tt'} X_{t'}^{n'} \right) + \sum_{t \in TE^{n+1}} \beta_t \left(s_t - \sum_{t' \in TS_t \cap TC^{n+1}} a_{tt'} X_{t'} \right)$$

$$s.t. \quad \sum_{t':t \in TS_{t'}} \beta_{t'} a_{t't} \leq h_t \quad \forall t \in TC^{n+1} \qquad \text{(BCn+1.1)}$$

$$\sum_{t':t \in TS_{t'}} \beta_{t'} a_{t't} \leq h_t \quad \forall t \in TE^{n+1}. \qquad \text{(BCn+1.2)}$$

Because of the cascade overlap, $TE^n \neq TC^{n+1} \cup TE^{n+1}$. Consequently, the subproblem extreme points required by $BCAS^n$ are defined by $BCn.1$, while the subproblem extreme points of $BCAS^{n+1}$ are defined by $BCn+1.1$ and $BCn+1.2$. Thus, the extreme points $\mathrm{b}^{(i)} \in \mathbf{B}$ required by $BCAS^n$ are not the same as the extreme points provided by the dual of $BCAS^{n+1}$. However, only the cascade overlap distinguishes the two feasible regions, which suggests the two regions share many similarities.

The first series of the cascading variation of Benders' decomposition solves subproblems $BCAS^1$,$BCAS^2$, ...,$BCAS^N$, without any cuts. Subsequent series solve these subproblems in the same order using the heuristic cuts generated by the dual variables from $BCn+1.1$ and $BCn+1.2$ of all previous series. Each series includes one additional cut per subproblem. The proximal cascade solution value is the objective value of the last subproblem of the most recent series, since $TE^N \neq \emptyset$. A Lagrangian cascade uses the dual variables supplied by the most recent proximal cascade.

Table 8 gives results for test sets 1 through 10. In general, the method does not converge to monolith optimal, but stabilizes consistently within a few percent in all of these examples. As with iterated Lagrange multipliers and forward pass multipliers, over half (60%) of the gap reduction is attributable to the Lagrangian cascade, which reflects the benefit of more accurate Lagrangian penalties.

These results suggest a promising alternative to Lagrangian methods of passing dual information within the proximal cascade (although this method should not be used when enforcing myopia). Unlike traditional nested decomposition for staircase models, the cascad-

Set #	Benders' Series until Stable	Unaltered % gap	Iterated Lagrange Multipliers % gap	Benders' % gap
1	13	7.0	.2	0
2	10	10.7	10.7	7.2
3	2	11.5	0	1.7
4	9	23.2	23.2	6.5
5	3	22.4	15.2	4.0
6	4	21.5	1.0	3.3
7	4	21.2	0	3.3
8	3	13.2	8.8	1.3
9	2	13.8	0	0
10	3	5.7	0	0

Table 8. Sets of cascade $BCAS$ use the same widths and overlaps as in Table 6 to test the cascading variation of Benders' decomposition. Each Benders' series includes one more dual cut in the proximal cascade than the previous series. By retaining old cuts, this method is more effective at gap reduction than unaltered or iterated Lagrange multipliers methods. For example, set #8 requires 3 Benders' series to stabilize at a gap of 1.3%. In contrast, the unaltered cascade produces a gap of 13.2%, while the iterated Lagrange multipliers cascade has a gap of 8.8%. The Benders' gap is always less than the unaltered gap, but is slightly larger than the iterated Lagrange multipliers gap in sets 3, 6, and 7. The average Benders' gap is 2.7%, while the unaltered and iterated Benders' gaps are 15.0% and 5.9%, respectively. Additionally, the Benders' approach typically yields gaps within a few percent, while the iterated Lagrange multipliers gaps are more erratic.

ing variation of Benders' lacks a convergence proof. However, traditional nested decompositions have no cascade overlap, and must enforce rows with fixed column levels at the risk of infeasibility. Thus, the cascading variation of Benders' decomposition has an advantage over many nested formulations, which must add cuts until convergence is obtained.

E. CASCADES WITH FIXED FUTURE PRIMALS

We improve solution quality on test problems by fixing all inactive columns at their last computed level, either from a (previous) proximal cascade subproblem of the current series, or from a proximal cascade subproblem of the previous series. The *fixed future primals* method is unlike other cascade strategies in this research because those strategies all fix future columns at zero. Fixing future primal columns based on previous series is not myopic, and it may offer considerable improvement in solution quality.

Primal variables indexed by $t > lastp^n$ do not influence cascade subproblem $SCAS^n$, since a row's cascade index is never exceeded by any associated column's cascade index. In

order to incorporate an influence of future columns, we define set $TU^n = \{t : lastp^n < t \leq lastp^n + m\}$ as the set of future periods just outside the active index set. Now consider subproblem SF^n which activates all rows associated with any active column:

$$(SF^n) \quad Z^n = \sum_{n'<n} \sum_{t \in TF^{n'}} h_t X_t^{n'} + \min \sum_{t \in TC^n} h_t X_t$$

$$s.t. \sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \geq s_t - \sum_{n'<n} \sum_{t' \in TS_t \cap TF^{n'}} a_{tt'} X_{t'}^{n'} \quad \forall t \in TC^n \qquad (SF1)$$

$$\sum_{t' \in TS_t \cap TC^n} a_{tt'} X_{t'} \geq s_t - \sum_{t' \in TS_t \cap TU^n} a_{tt'} \widehat{X}_{t'} \quad \forall t \in TU^n \qquad (SF2)$$

$$X_t \geq 0 \qquad \forall t \in TC^n.$$

Here, $\widehat{X}_t$ is the last value computed for a column from a cascade of the previous series. $\widehat{X}_t = 0 \ \forall t$ in the first series.

The fixed future primals method is appealing because it may increase solution quality with only a minor formulation change, but this change may have disadvantages. For example, including constraint *SF2* in the formulation may produce infeasibility when the constraints represent balance of flows, particularly in the first series when $\widehat{X}_t = 0$. This difficulty motivated the discussion in Chapter II, Section B.2, and precipitated leaving these rows inactive for other cascade descriptions.

Constraint *SF2* also increases the number of rows in each subproblem without guaranteeing improvement of the solution quality. Consider this simple example:

$$\begin{array}{lrrrrr} \min & 4X_1 & +2X_2 & +X_3 & & \\ s.t. & 10X_1 & +X_2 & & \geq & 2 \\ & & +X_2 & X_3 & \geq & 2 \\ & X_1, & X_2, & X_3 & \geq & 0. \end{array}$$

A solution to this problem is: $X_1^* = .2, \ X_3^* = 2$, with $Z^* = 2.8$. Now consider a cascade

with two subproblems:

$$\begin{array}{llllll} \min & 4X_1 & +2X_2 & & & \\ s.t. & 10X_1 & +X_2 & \geq & 2 & \\ & & X_2 & \geq & 2 & \\ & X_1, & X_2, & \geq & 0 & \end{array} \qquad X_1^1 + \begin{array}{llllll} \min & 2X_2 & +X_3 & & & \\ s.t. & X_2 & & \geq & 2-10X_1^1 & \\ & X_2 & +X_3 & \geq & 2 & \\ & X_2, & X_3 & \geq & 0 & \end{array}$$

$$\begin{array}{ll} Z^1 = & 4 \\ X_1^1 = & 0 \\ X_2^1 = & 2 \end{array} \qquad \begin{array}{ll} Z^2 = & 4 \\ X_2^2 = & 2 \\ X_3^2 = & 0. \end{array}$$

Since $X_3^2 = 0$, subsequent series will generate the same solution and no improvement in solution quality can result.

Although it does not guarantee better solution quality, the fixed future primals method works well in test sets. Table 9 compares fixed future primals with the Benders' method of the last section. The two methods offer similar results, although the Benders' method consistently yields gaps under 10%, while fixed future primals gaps are more erratic.

Fixed future primals offer another method to improve solution quality when myopia is not required. As with iterated dual multipliers and the cascading variation of Benders' decomposition, fixed future primals require multiple cascade series. This method is simpler to implement than adding Benders' cuts, although subproblem infeasibility is more likely due to the increased number of constraints with inactive associated columns. The resulting gaps are smaller than with the Lagrangian penalty method, but larger than the gaps obtained with the Benders' method.

F. SUMMARY

Cascades often provide a useful alternative to actually solving a linear programming monolith. Linear programs with constraints that can be re-ordered into a staircase are usually suitable for cascading. The *casfactor*$_T$, w_T, and *allact*$_T$ gauges provide an indication of cascade suitability, although some formulations complicate their effectiveness.

A suitable model warrants cascading if the monolith is too large to solve, or if the model requires myopic solution, or if we can isolate "easy" subproblems that facilitate rapid

Set #	Benders' % gap	Primal Series 1 % gap	Primal Series 2 % gap	Primal Series 3 % gap	Primal Series 4 % gap	Primal Series 5 % gap	Primal Series 6 % gap
1	0	7.0	.6	0			
2	7.2	12.9	7.4	5.1	4.0	2.7	-
3	1.7	10.6	7.8	-	-	-	-
4	6.5	8.2	4.7	-	-	-	-
5	4.0	18.5	12.5	-	-	-	-
6	3.3	27.7	.3	0			
7	3.3	46.0	4.6	3.3	-	-	-
8	1.3	5.5	5.5	2.3	1.3	0	
9	0	13.8	12.4	-	-	-	-
10	0	5.6	1.3	-	-	-	-

Table 9. Sets of cascade SF use various widths and overlaps to compare *fixed future primals* with the cascading variation of Benders' decomposition. In the primal method, each proximal cascade subproblem incorporates columns fixed in the previous series to estimate future columns in the current series, therereby reducing myopia. We terminate each set when a series yields a gap of zero, or when no further gap reduction occurs. For instance, Set #6 has a Benders' gap of 3.3%. The first fixed future primals series uses future column levels fixed at 0, and yields a gap of 27.7%. Subsequent fixed future primals gaps for set #6 are .3%, and 0%. A "-" series entry indicates that no further gap improvement occurs. Fixed future primals yield better solution quality than Lagrangian penalty methods, although the solution qualties are not, on average, as good as the Benders' solution qualities. The average fixed future primals gap is 4.5%, and the average Benders' gap is 2.7%.

monolith solution. Cascades are a viable alternative to attempting an outright monolith solution in any of these situations.

Finally, incorporating dual, or additional primal information in a proximal cascade reduces the proximal-Lagrangian gap. Forward pass multipliers preserve myopia by incorporating dual information from structurally similar constraints of previous subproblems. Other methods require multiple cascade series. The iterated Lagrange multipliers method, fixed future primals method, and the cascading variation of Benders' decomposition all violate myopia, but provide more gap reduction than a myopic cascade.

VI. SUMMARY AND RECOMMENDATIONS

This dissertation develops a proximal cascade heuristic to approximate solutions of large linear programs, and a Lagrangian cascade to bound the error incurred by that approximation. NRMO demonstrates the usefulness of cascades on a large, time-based linear program. This model has the staircase structure conducive to cascades, but also includes numerous complexities that allow us to illustrate the flexibility of cascades. This dissertation also examines the applicability of cascades to more general models, and presents enhancements that reduce the proximal-Lagrangian gap.

A. CONTRIBUTIONS

1. Large-Scale Mathematical Programming

There are several contributions to the solution techniques for large-scale optimization models presented in this research. First, we formalize the proximal cascade heuristic, and present results that relate the cascade solution value to the monolith solution value for staircase linear programs. Although we present the results in the context of elastic-demand staircase LPs, they are directly applicable to more general staircase models. Additionally, we develop optimistic bounds on the monolith solution value using a proximal cascade for a specific class of problems.

This research also introduces and develops the Lagrangian cascade. Using dual prices from the proximal cascade, a Lagrangian cascade serves as a super-optimal bound on the monolith solution to any staircase model. Previously, no technique existed to evaluate how close the proximal cascade solution value might be to the monolith solution value. We strengthen the Lagrangian cascade bound by the introduction of extended constraints and duplicate variables. This technique is generalizable to any Lagrangian relaxation.

We extend the theory of proximal cascades to utilize dual as well as primal variables in subsequent cascades or subproblems. The fixed future primals method improves

solution quality by fixing the future columns of the subproblems at their last computed values. Dual information can improve the cascade solution quality by rewarding satisfaction of constraints that are explicitly enforced in later subproblems. There are three specific dual techniques, including: 1) iterated Lagrange multipliers, which use duals from the corresponding constraints of previous cascade series, 2) forward pass multipliers, which forecast future dual prices using a previous subproblem, and 3) a cascading variation of Benders' decomposition, which adds successive dual cuts to each subproblem.

2. Air Mobility Optimization

This research describes a large-scale optimization model to enhance DOD's mobility analysis capability. NRMO, jointly developed with Rosenthal, Morton, and Melody, is the most detailed optimization model of USAF air mobility assets ever developed. It is currently in use by the RAND Corporation, has been accepted by the Air Force Studies and Analyses Agency, and is being evaluated by Headquarters, Air Mobility Command for use in forthcoming studies.

Cascades alleviate the primary concern in the mobility analysis community that an instance of a NRMO model, large enough to capture adequate detail, is too large to solve. Cascading reduces NRMO solution times by as much as 80% when available memory is limited. Cascades also eliminate the secondary concern that optimizing the air mobility system erroneously assumes perfect scheduling foresight, although the actual system is myopic. Furthermore, this dissertation demonstrates the availability of optimistic bounds that are used to examine the cost of myopic scheduling.

B. RECOMMENDATIONS FOR FUTURE RESEARCH

We suggest some interesting enhancements to cascades for further study:

- A cascade of integer programs is mentioned in this research but not developed. Although the proximal cascade results apply directly to integer programs, the Lagrangian cascade results do not. Lagrangian relaxations of integer programs may exhibit an integrality gap, since even optimal dual multipliers do not necessarily provide a tight bound. Non-optimal multipliers from a proximal cascade will generally degrade this gap even further. Additionally, a Lagrangian cascade

requires meaningful dual information from the proximal cascade. Meaningful duals may not be available when the proximal cascade includes integer restrictions.

- A cascade subproblem's first and last periods in this research are set at regular intervals. Model structure may suggest alternate "boundaries" where the model is weakly linked. Exploiting this information *a priori* could increase cascade solution quality, but appears to be very model specific and difficult to predict in general.
- Penalties taken from previous proximal cascade subproblems and applied to the active subproblem are approximated crudely in this research. When possible, penalties should be chosen to exploit suspected similarities between previously constrained resources and resources of the active subproblem. Simple techniques such as exponential smoothing could also have merit.
- Chapter V does not develop any explicit modifications of the Lagrangian cascade to reduce the proximal-Lagrangian cascade gap. The use of well-documented multiplier search methods on a cascade could significantly reduce this gap, although the computational burden would be increased.

C. CONCLUSION

Cascades provide a useful approximation strategy when problem structure permits, and when model size or system myopia warrants. This research formalizes a proximal cascade approximation on a class of problems, develops a Lagrangian cascade bound on that approximation, and demonstrates the combined approach on a model currently in use by the USAF.

LIST OF REFERENCES

Ahuja, R., Magnanti, R., and Orlin, J., *Network Flows,* Prentice Hall, Englewood Cliffs NJ, 1993.

Aronson, J., Morton, T., and Thompson, G., "A Forward Simplex Method for Staircase Linear Programs," *Management Science* 31 (1985), 664-679.

Bazaraa, M., and Sherali, H., and Shetty, C., *Nonlinear Programming,* John Wiley & Sons, New York, 1993.

Benders, J., "Partitioning Procedures for Solving Mixed Variables Programming Problems," *Numerische Mathematik* 4 (1962), 238-252.

Birge, J., "Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs," *Operations Research* 33 (1985), 989-1007.

Brooke, A., Kendrick, D., and Meerhaus, A., *GAMS, a User's Guide*, The Scientific Press, South San Francisco, 1992.

Brown, G., Dell, R., and Wood, R., "Optimization and Persistence," *Interfaces* (forthcoming).

Brown, G., Graves, G., and Honczarenko, M., "Design and Operation of a Multicommodity Production/Distribution System Using Primal Goal Decomposition," *Management Science* 33 (1987), 1469-1480.

Brown, G., and Graves, G, and Ronen, D., "Scheduling Ocean Transportation of Crude Oil," *Management Science* 33 (1987), 335-346.

Charnes, A., and Cooper, W., *Management Models and Industrial Applications of Linear Programming*, John Wiley & Sons, New York, 1961.

CPLEX Optimization Inc., *CPLEX,* version 3.0 callable library, Incline Village NV, 1994.

Dantzig, G., and Fulkerson, D., "Minimizing the Number of Tankers to Meet a Fixed Schedule," *Naval Research Logistics Quarterly* 1 (1954), 217-222.

Dantzig, G., and Wolfe, P., "Decomposition Principle for Linear Programs," *Operations Research* 8 (1960), 101-111.

Department of the Air Force, *Basic Aerospace Doctrine of the United States Air Force,* Air Force Manual 1-1, Vol. II, US Government Printing Office, 1992.

Geoffrion, A., and Graves, G., "Multicommodity Distribution System Design by Benders Decomposition," *Management Science* 20 (1974), 822-844.

Glassey, C., "Nested Decomposition and Multistage Linear Programs," *Management Science* 20 (1973), 282-292.

Grinold, R., "Model Building Techniques for the Correction of End Effects in Multi-Stage Convex Programs," *Operations Research* 31 (1983), 407-431.

Hillier, F., and Lieberman, G., *Introduction to Operations Research*, Holden-Day, Inc., Oakland CA, 1986.

Ho, J., and Manne, A. S., "Nested Decomposition for Dynamic Models," *Mathematical Programming* 6 (1974), 121-140.

Jayakumar, M., and Ramasesh, R., "A Solution-Cascading Approach to the Decomposition of Staircase Linear Programs," *Journal of the Operational Research Society* 3 (1994), 301-308.

Johnson, L., and Montgomery, P., *Operations Research in Production Planning, Scheduling, and Inventory Control*, John Wiley & Sons, New York, 1974.

Killingsworth, P., and Melody, L., "CONOP Air Mobility Optimization Model," RAND, Santa Monica, CA, 1994.

Klotz, E., "Using *CPLEX* and the *CPLEX* Callable Library," training class, *CPLEX* Optimization Inc., Incline Village, NV, October, 1996.

Lim, T., *Strategic Airlift Assets Optimization Model,* Master's Thesis, Operations Research Department, Naval Postgraduate School, CA, 1994.

Manne, A., "Sufficient Conditions for Optimality in an Infinite Horizon Development Plan," *Econometrica* 38 (1970), 18-38.

Morton, D., Rosenthal, R., and Lim, T., "Optimization Modeling for Airlift Mobility," *Military Operations Research* 1:4, (1996), 49-67.

Morton, T., "Forward Algorithms for Forward Thinking Managers," *Applications of Management Science,* Vol. 1, R. Schultz (Ed), JAI Press Inc., 1981.

Parker, R., and Rardin, R., *Discrete Optimization*, Academic Press, Boston, 1988.

Rosenthal, R., Baker, S., Lim, T., Fuller, D., Goggins, D., Toy, A., Turker, Y., Horton, D., Briand, D., and Morton, D., *Application and Extension of the THRUPUT II Optimization Model For Airlift Mobility*, NPS-OR-96-018, Naval Postgraduate School , Monterey, CA, 1996.

Rosenthal, R., Morton, D., Melody., L, and Baker, S., "NPS/RAND Mobility Optimizer," internal memorandum, Naval Postgraduate School, Monterey, CA, 1997.

Staniec, C., *Solving the Multicommodity Transshipment Problem*, Ph.D. Dissertation, Operations Research Department, Naval Postgraduate School, CA, 1987.

Stucker, J., and Melody, L., "Impacts of European Airbase Infrastructure on Airlift Mobility," RAND study for the Office of the Secretary of Defense, 1996.

Van Slyke, R., and Wets, R., "L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Linear Programs," *SIAM Journal on Applied Mathematics* 17 (1969), 638-663

Walker, S., *Evaluating End Effects for Linear and Integer Programs Using Infinite-Horizon Linear Programming*, Ph.D. Dissertation, Operations Research Department, Naval Postgraduate School, CA, 1995.

Walker, S., and Dell, R., "Quantifying End Effects in Linear and Integer Programs Using Infinite Horizon Programming," Naval Postgraduate School, Monterey, CA, October 1995.

Webster's Third New International Dictionary of the English Language, Merriam-Webster, Springfield MA., 1993.

Wing, V., Rice, R., Sherwood, R., and Rosenthal., R., "Determining the Optimal Mobility Mix," Joint Staff (J8), Force Design Division, The Pentagon, Washington, DC, 1 October 1991.

Yost, K., "The THRUPUT Strategic Airlift Flow Optimization Model, " internal memorandum, Air Force Studies and Analyses Agency, 30 June 1994.

Zipkin, P., "Bounds for Row Aggregation in Linear Programming," *Operations Research* 28 (1980), 903-916.

INITIAL DISTRIBUTION LIST

No. of Copies

1. Defense Technical Information Center 2
   8725 John J. Klingman Rd., Ste 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Dr. Richard S. Elster Code 01 1
   Naval Postgraduate School
   Monterey, CA 93943-5002

4. Dr. Peter Purdue Code 08 1
   Naval Postgraduate School
   Monterey, CA 93943-5002

5. Dr. Frank C. Petho Code OR/Pe 1
   Naval Postgraduate School
   Monterey, CA 93943-5002

6. Dr. Gerald G. Brown Code OR/Bw 1
   Naval Postgraduate School
   Monterey, CA 93943-5002

7. Dr. David P. Morton 1
   Department of Mechanical Engineering
   Engineering Teaching Center
   The University of Texas at Austin
   Austin, TX 78712

8. Dr. Craig M. Rasmussen Code MA/Ra 1
   Naval Postgraduate School
   Monterey, CA 93943-5002

9. Dr. R. Kevin Wood Code OR/Wd 1
   Naval Postgraduate School
   Monterey, CA 93943-5002

10. Dr. Richard E. Rosenthal Code OR/Rl 10
    Naval Postgraduate School
    Monterey, CA 93943-5002

11. Major Steven F. Baker 10
    Department of Management
    2354 Fairchild Dr. Ste. 6H94
    USAF Academy, CO 80840-5701

12. AFSAA/SAGM 3
    1570 Airforce Pentagon
    Washington DC, 20330-1570

13. General Walter Kross 1
    AMC/CC
    402 Scott Dr. Unit 3EC
    Scott AFB IL, 62225-5307

14. HQ AMCSAF/XPY 3
    Attn: Col. Michael Baum, Capt. Jean Steppe, Mr. Alan Whisman
    402 Scott Dr. Unit 3L3
    Scott AFB IL, 62225-5307

15. AFIT/CIGS Bldg 125 1
    2950 P. Street
    Wright-Patterson AFB, OH 45433-7765

16. AFIT/ENS Bldg 640 1
    Attn: Ltc. James Moore
    2950 P. Street
    Wright-Patterson AFB, OH 45433-7765

17. AFOSR/NM 1
    Attn: Dr. Neal Glassman
    110 Duncan Av, Ste. 100
    Bolling AFB, Washington DC 20332-0001

18. Air University Library (AU/LD) 1
    600 Chennault Cir.
    Maxwell AFB, AL 36112-5001